

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ



Εφαρμογή σε πλατφόρμα Android για τον εντοπισμό σταθμευμένου οχήματος

An Android car locator

Διπλωματική Εργασία

Καλφούντζος Δημήτριος

Υπεύθυνος καθηγητής: Μποζάνης Παναγιώτης

Καθηγητής Π.Θ

Επιβλέπων καθηγήτρια: Τσομπανοπούλου Παναγιώτα

Επίκουρος Καθηγήτρια Π.Θ

Βόλος, 2015

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

**Εφαρμογή σε παλτφόρμα Android για τον εντοπισμό
σταθμευμένου οχήματος**

Διπλωματική Εργασία

Καλφούντζος Δημήτριος

Υπεύθυνος καθηγητής: Μποζάνης Παναγιώτης

Καθηγητής Π.Θ

Επιβλέπων καθηγήτρια: Τσομπανοπούλου Παναγιώτα

Επίκουρος Καθηγήτρια Π.Θ

Εγκρίθηκε από την Διμελή Εξεταστική επιτροπή τον Ιούνιο του 2015.

Υπογραφή

Υπογραφή

(.....)

Μποζάνης Παναγιώτης

Αναπληρωτής Καθηγητής Π.Θ

(.....)

Τσομπανοπούλου Παναγιώτα

Επίκουρος Καθηγήτρια Π.Θ

Ευχαριστίες

Στα πλαίσια της διπλωματικής μου εργασίας θα ήθελα να ευχαριστήσω όλους όσους στάθηκαν δίπλα μου τόσο σε ακαδημαϊκό όσο και σε προσωπικό επίπεδο.

Πάνω από όλα θα ήθελα να πω ένα μεγάλο ευχαριστώ στην οικογένεια μου που ήταν δίπλα μου και μου στάθηκε όλα αυτά τα χρόνια με κάθε εφικτό τρόπο.

Στη συνέχεια θα ήθελα να ευχαριστήσω θερμά τον κύριο Μποζάνη Παναγιώτη -καθηγητή και επιβλεποντα της πτυχιακής- για την καθοδήγηση του, και την πολύτιμη στήριξη του, καθώς επίσης και την κυρία Τσομπανοπούλου Παναγιώτα –επίκουρο καθηγήτρια και συνεπιβλέπουσα της πτυχιακής.

Επίσης θα ήθελα να ευχαριστήσω όλους τους φίλους μου για την υποστήριξη και τη βοήθεια που παρείχαν προς το προσώπό μου.

Τέλος μέσα από την καρδιά μου θα ήθελα να πω ένα μεγάλο ευχαριστώ σε ένα πρόσωπο κλειδί για την για την ακαδημαϊκή μου πορεία την Κατερίνα. Την ευχαριστώ για την βοήθειά της , τις συμβουλές της και την καθοδήγηση της.

Καλφούντζος Δημήτριος

Λάρισα, 2015

Περίληψη

Η παρούσα διπλωματική εργασία “Ανάπτυξη εφαρμογής εντοπισμού σταθμευμένου οχήματος σε πλατφόρμα Android” έχει στόχο την υλοποίηση εφαρμογής σε περιβάλλον Android.

Η υλοποίηση της πραγματοποιήθηκε με το λογισμικό ελεύθερης διανομής (freeware) και ανοικτού κώδικα(open source) Eclipse – Juno.

Η εφαρμογή μπορεί να χρησιμοποιηθεί από όλους τους χρήστες κινητών τηλεφώνων με λειτουργικό Android.Ο χρήστης της εφαρμογής θα έχει τη δυνατότητα να αποθηκεύει τη θέση στάθμευσης του οχήματός του με αποτέλεσμα από οποιοδήποτε σημείο βρίσκεται ανά πάσα στιγμή, να καθοδηγείται στο όχημα του μέσω της διαδρομής που θα του εμφανίζεται στον χάρτη(Google Maps).

Η συγκεκριμένη εφαρμογή απευθύνεται σε χρήστες οι οποίοι κινούνται σε πόλεις και οδικά δίκτυα τα οποία δεν τους είναι οικεία. Άρα μέσω της χρήσης της δεν θα είναι αναγκαίο να απομνημονεύουν τη διαδρομή ώστε να ξαναβρεθούν στο σταθμευμένο όχημα τους.

ΠΕΡΙΕΧΟΜΕΝΑ

1 ΕΙΣΑΓΩΓΗ.....	8
1.1 Έξυπνα τηλέφωνα.....	8
1.2 Λειτουργικό σύστημα	9
1.2.1 Λειτουργικά συστήματα για έξυπνα τηλέφωνα	9
1.3 Τι είναι το Android;	10
1.4 Εφαρμογές Android	11
1.5 Εξέλιξη του Android	12
1.5.1 Android 1.5 Cupcake (API Level: 3).....	12
1.5.2 Android 1.6 Donut (API Level: 4)	13
1.5.3 Android 2.0/2.1 Eclair (API Level: 7)	13
1.5.4 Android 2.2 Froyo (API Level: 8).....	14
1.5.5 Android 2.3 Gingerbread (API Level: 9)	14
1.5.6 Android 3.0 Honeycomb (API Level: 13).....	15
.....	16
1.5.7 Android 4.0 Ice cream sandwich (API Level: 15).....	16
1.5.8 Android 4.1 Jelly Bean (API Level: 16)	16
1.5.9 Android 4.4 Kit Kat (API Level: 19)	17
2 Αρχιτεκτονική Android	18
2.1 Πυρήνας Linux (Linux Kernel)	19
2.2 Βιβλιοθήκες	19
2.3 Η εικονική μηχανή Dalvik.....	20
2.4 Χρόνος Εκτέλεσης Εφαρμογής (Android Runtime)	20
2.5 Πλαίσιο Εφαρμογής (Application Framework)	21
2.6 Εφαρμογές (Applications)	21
3 Εργαλεία ανάπτυξης της εφαρμογής	22
3.1 Android SDK	23
3.2 Android NDK.....	24
3.3 Eclipse	24
3.4 Android Development Tools (ADT)	25

3.5 Εγκατάσταση του Eclipse IDE	25
3.6 Εγκατάσταση του Android SDK Manager	27
3.7 Δημιουργία νέου Project	28
4 Δομή της εφαρμογής Android Locator.....	29
4.1 Η κλάση AsyncTask.....	29
4.2 Android Manifest	30
4.3 Activity	32
4.3.1 Fragments.....	33
4.4 Resources.....	34
4.5 Services	34
4.6 Android SQLite Database.....	35
4.7 Content providers	35
4.8 Διεπαφή χρήστη	36
4.8.1 Layout	36
4.8.2 Menu	37
4.8.3 Dialogs.....	38
4.8.4 Notifications	39
4.9 Google Maps.....	39
5 Υλοποίηση και λειτουργικότητα της εφαρμογής	40
5.1 Περιγραφή της εφαρμογής	40
5.2 Δοκιμές εφαρμογής.....	44
5.3 Εξέλιξη της εφαρμογής.....	45
ΠΑΡΑΡΤΗΜΑ	46
ΕΥΡΗΤΗΡΙΟ ΕΙΚΟΝΩΝ.....	50
ΑΝΑΦΟΡΕΣ – ΒΙΒΛΙΟΓΡΑΦΙΑ.....	52

1 ΕΙΣΑΓΩΓΗ

1.1 Έξυπνα τηλέφωνα

Τα έξυπνα τηλέφωνα είναι κινητά τηλέφωνα βασισμένα σε ένα λειτουργικό σύστημα με προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα σε σχέση με τα απλά κινητά τηλέφωνα. Τα πρώτα smartphones εκτός από τις λειτουργίες ενός κινητού διέθεταν επιπλέον τις λειτουργίες ενός PDA(Προσωπικός Ψηφιακός Οδηγός). Αργότερα προστέθηκαν λειτουργίες των φορητών media players, low-end compact ψηφιακές φωτογραφικές μηχανές, βιντεοκάμερες τσέπης, καθώς και μονάδες πλοήγησης GPS.

Πλέον τα σύγχρονα smartphones περιλαμβάνουν οθόνες αφής υψηλής ανάλυσης και διαδικτυακούς περιηγητές (web browsers) που εμφανίζουν τυποποιημένες ιστοσελίδες καθώς και βελτιστοποιημένες ιστοσελίδες για κινητά. Η πρόσβαση των smartphones στο διαδίκτυο γίνεται μέσω WI-FI.[1]



Εικόνα 1: Smartphones(ti-einai.gr) [1]

1.2 Λειτουργικό σύστημα

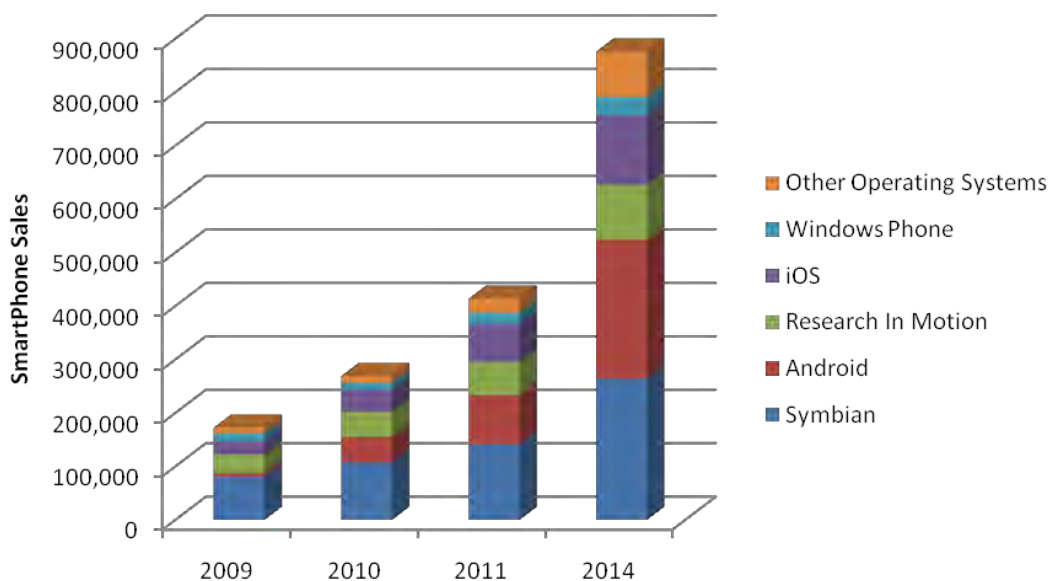
Λειτουργικό σύστημα στην επιστήμη της πληροφορικής είναι το λογισμικό του υπολογιστή που είναι υπεύθυνο για τη διαχείριση και τον συντονισμό των εργασιών και την κατανομή των διαθέσιμων πόρων. Το λειτουργικό σύστημα παρέχει ένα θεμέλιο, ένα μεσολαβητικό επίπεδο λογικής διασύνδεσης μεταξύ λογισμικού και υλικού, διαμέσου του οποίου οι εφαρμογές αντιλαμβάνονται εμμέσως τον υπολογιστή. Μια από τις κεντρικές αρμοδιότητες του λειτουργικού συστήματος είναι η διαχείριση του υλικού, απαλλάσσοντας έτσι το λογισμικό του χρήστη από τον άμεσο και επίπονο χειρισμό του υπολογιστή και καθιστώντας ευκολότερο τον προγραμματισμό τους.



Εικόνα 2: Τυπική θέση του λειτουργικού συστήματος σε ένα υπολογιστικό σύστημα(wiki)[2]

1.2.1 Λειτουργικά συστήματα για έξυπνα τηλέφωνα

Τα λειτουργικά συστήματα (OS) των κινητών τηλεφώνων που χρησιμοποιούνται από τα σύγχρονα smartphones περιλαμβάνουν το Android της Google, το iOS της Apple, το Symbian της Nokia, το BlackBerry OS της RIM, το Bada της Samsung, τα Windows Phone της Microsoft, το webOS της Hewlett-Packard, καθώς και ενσωματωμένες διανομές Linux όπως το Maemo και το MeeGo. Τα πιο διαδεδομένα βέβαια είναι αυτά της Google και της Apple, όπως φαίνεται και στην Εικόνα 3, καθώς κατέχουν την μερίδα του λέοντος της παγκόσμιας αγοράς.



Εικόνα 3: Smartphone Sales by OS(blogs.amdocs.com) [3]

1.3 Τι είναι το Android;

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Στο android, μπορούμε να εγκαταστήσουμε εφαρμογές (όπως επεξεργαστές κειμένου, φωτογραφίας, αναπαραγωγή μουσικής, παιχνίδια, κλπ κλπ) να συνδεθούμε σε δίκτυα και να περιηγηθούμε στο internet και πολλά άλλα.

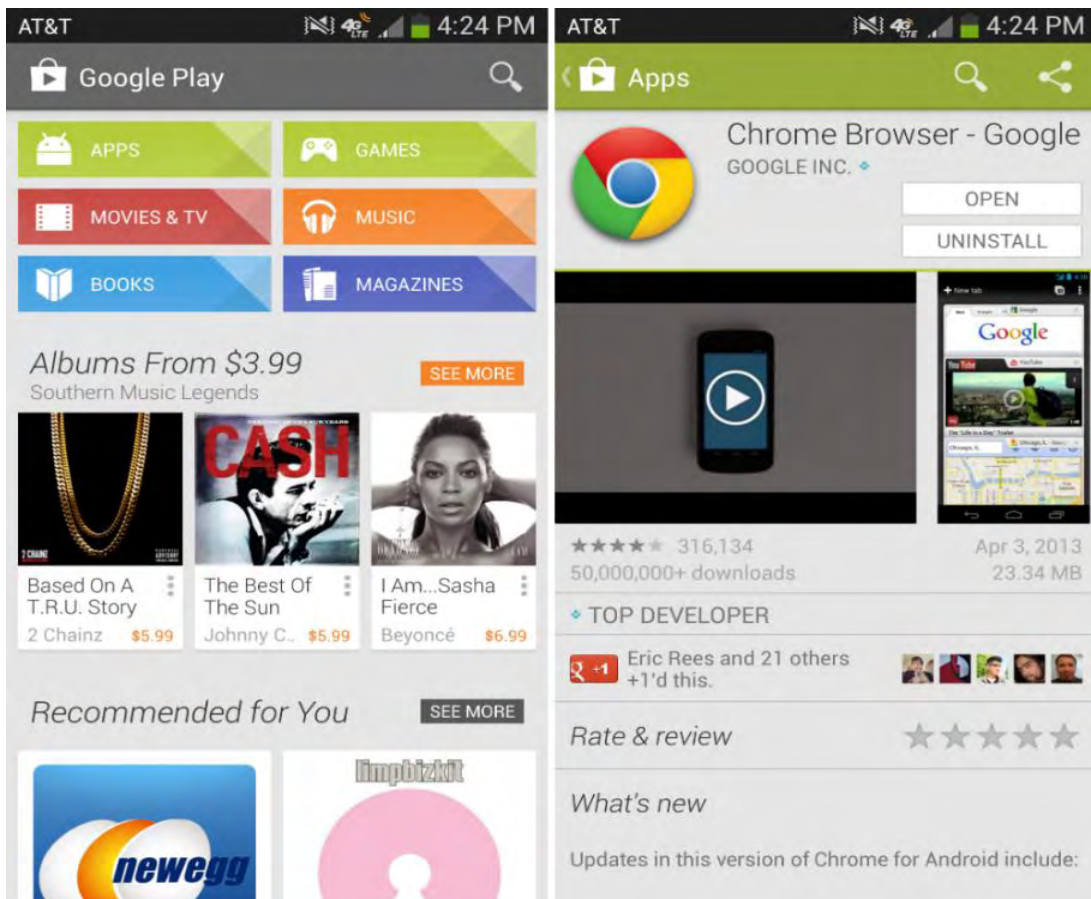
Το Android κατά κύριο λόγο είναι σχεδιασμένο ώστε να εκμεταλλεύεται τους πόρους ενός κινητού τηλεφώνου - smartphone ή tablet, και είναι αποκλειστικά σχεδιασμένο για αυτά (βάσει των ιδιοτήτων τους και της διαφορετικότητάς τους από τους κοινούς υπολογιστές) κι όχι για υπολογιστή desktop ή laptop.

Γ' αυτό και τα προγράμματα και οι εφαρμογές που εγκαθιστούμε σε αυτό, θα πρέπει να είναι για android ή να υποστηρίζονται έστω από αυτό το λειτουργικό σύστημα.

Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου.[2][5]

1.4 Εφαρμογές Android

Το Android έχει μια μεγάλη κοινότητα προγραμματιστών που γράφουν εφαρμογές, οι οποίες επεκτείνουν τη λειτουργικότητα των συσκευών. Οι εφαρμογές γράφονται σε μια προσαρμοσμένη έκδοση της JAVA και μπορεί κανείς να τις κατεβάσει από το online κατάστημα Google Play (πρώην Android Market) της Google όπως και από άλλα sites. Το Android Market, είναι ένα online κατάστημα της Google, προσφέρει σε κάθε χρήστη εφαρμογές για το κινητό του που είναι συμβατό με το λειτουργικό της Google. Το συντριπτικό ποσοστό των εφαρμογών είναι δωρεάν ενώ πλέον οι πληρωμένες εφαρμογές είναι διαθέσιμες και στο ελληνικό κοινό. Μέχρι τον Ιούνιο του 2014 περισσότερες από 1,2 εκατομμύρια εφαρμογές ήταν διαθέσιμες για Android ενώ ο αριθμός των downloads από το Google Play μέχρι το Ιούλιο του 2013 είχε υπερβεί τα 50 δισεκατομμύρια.[5]



Εικόνα 4: Google Play(droid-life.com) [4]

1.5 Εξέλιξη του Android

Η ιστορία των εκδόσεων του Android λειτουργικού συστήματος για τα κινητά τηλέφωνα ξεκίνησε με την κυκλοφορία του Android beta το Νοέμβριο του 2007. Η πρώτη εμπορική έκδοση ήταν το Android 1.0 που κυκλοφόρησε το Σεπτέμβριο του 2008. Το Android είναι υπό συνεχή ανάπτυξη από την Google και την Open Handset Alliance (OHA), έχοντας διαθέσει στο κοινό μία σειρά από ενημερώσεις στην λειτουργία του συστήματος από την αρχική κυκλοφορία του.

Από τον Απρίλιο του 2009, οι εκδόσεις του Android επηρεάστηκαν από την ζαχαροπλαστική για την ονομασία τους εξαιρουμένων των εκδόσεων 1.0 και 1.1, που δεν τέθηκαν υπό συγκεκριμένα ονόματα:

- Apple Pie (1.0)
- Banana Bread (1,1)
- Cupcake (1,5)
- Donut (1,6)
- Eclair (2.0-2.1)
- Froyo (2.2-2.2.3)
- Gingerbread (2.3-2.3.7)
- Honeycomb (3.0-3.2.6)
- Ice Cream Sandwich (4.0-4.0.4)
- Jelly Bean (4.1-4.3.1)
- KitKat (4.4-4.4.4)
- Lollipop (5.0-5.0.2)

1.5.1 Android 1.5 Cupcake (API Level: 3)

Η έκδοση "Cupcake" παρουσιάστηκε στις 30 Απριλίου του 2009. Υποστηρίζει νέες λειτουργίες για την κάμερα τις συσκευής, όπως η καταγραφή και παρακολούθηση βίντεο από την λειτουργία της κάμερας και η άμεση μεταφόρτωση του βίντεο αλλά και των φωτογραφιών στο Youtube και το Picasa αντίστοιχα, απευθείας από το τηλέφωνο. Έχει νέο έξυπνο

πληκτρολόγιο με πρόβλεψη κειμένου. Υποστηρίζει πρότυπο Bluetooth A2DP και AVRCP ενώ έχει και την ικανότητα να συνδέεται αυτόματα σε μικροσυσκευές Bluetooth από μια συγκεκριμένη απόσταση. Ακόμα στην έκδοση αυτή έχει νέο γραφικό περιβάλλον με κινούμενες μεταβάσεις οθόνης.[3]



Εικόνα 5: Λογότυπο Cupcake(androidoriginal.blogspot.gr) [5]

1.5.2 Android 1.6 Donut (API Level: 4)

Η έκδοση "Donut" παρουσιάστηκε στις 15 Σεπτεμβρίου του 2009. Έχει ταχύτερη απόκριση σε σχέση με την προηγούμενη έκδοση. Υποστηρίζεται πλέον η επιλογή πολλαπλών αρχείων ταυτόχρονα, έχει ανανεωμένο γκάλερι και φωτογραφική μηχανή, καθώς και βελτιωμένο Android Market. Έχει ανανεωμένη φωνητική αναζήτηση, με ταχύτερη απόκριση και βαθύτερη ολοκλήρωση με εγγενείς (native) εφαρμογές, συμπεριλαμβανομένης της δυνατότητας κλήσης επαφών. Δυνατότητα αναζήτησης σελιδοδεικτών, ιστορικού, επαφών αλλά και στο διαδίκτυο από την αρχική οθόνη. Υποστήριξη για ανάλυση οθονών WVGA. Ανανεωμένη υποστήριξη τεχνολογιών για CDMA/EVDO, 802.1x, VPNs και με μηχανή μετατροπής κειμένου σε ομιλία (text-to-speech).[3]



Εικόνα 6: Λογότυπο Donut(logos.wikia.com) [6]

1.5.3 Android 2.0/2.1 Eclair (API Level: 7)

Η έκδοση "Eclair" παρουσιάστηκε στις 26 Οκτωβρίου του 2009, ενώ τον Ιανουάριο του 2010 επανεκδόθηκε σε Android 2.1 Eclair (MR1). Σε αυτή την

έκδοση υπάρχει ακόμα ταχύτερη απόκριση του υλικού σε σχέση με τις δυο προηγούμενες και πλέον υποστηρίζονται περισσότερες οθόνες και αναλύσεις. Υπάρχει νέος browser ο οποίος υποστηρίζει το πρότυπο HTML5, νέο User Interface, και βελτιωμένοι χάρτες Google (Google Maps 3.1.2). Έχει ενσωματωθεί η υποστήριξη φλας για την κάμερα η οποία έχει πλέον και ψηφιακό zoom. Επίσης έχει βελτιωθεί η κλάση MotionEvent ώστε να υπάρχει η δυνατότητα για γεγονότα πολλαπλής αφής (multitouch events). Υποστηρίζεται Bluetooth 2.1 και έχει βελτιωθεί και το πληκτρολόγιο. .[3]



Εικόνα 7: Λογότυπο Éclair(sites.google.com) [7]

1.5.4 Android 2.2 Froyo (API Level: 8)

Η έκδοση “Froyo” παρουσιάστηκε στις 20 Μαΐου του 2010. Υπάρχουν βελτιστοποιήσεις στην ταχύτητα γενικά του λειτουργικού συστήματος, στην μνήμη και στην απόδοση. Έχει ενσωματωθεί ο μηχανισμός JavaScript του Chrome V8 στον browser, υπάρχει πλέον Adobe Flash 10.1, ενώ υποστηρίζεται καλύτερα πλέον το Microsoft Exchange. Έχει γίνει ανανέωση του Android Market. Ο χρήστης μπορεί πλέον να ελέγχει αν θα γίνεται ή όχι κίνηση πακέτων δεδομένων από το δίκτυο κινητής τηλεφωνίας. Υπάρχει η δυνατότητα εγκατάστασης εφαρμογών στην κάρτα μνήμης και η μεταφορά τους εκεί από τη μνήμη του τηλεφώνου. Επίσης το τηλέφωνο πλέον μπορεί να μετατραπεί σε WiFi hotspot. .[3]



Εικόνα 8: Λογότυπο Froyo(computerworld.com) [8]

1.5.5 Android 2.3 Gingerbread (API Level: 9)

Η έκδοση “Gingerbread” παρουσιάστηκε στις 6 Δεκεμβρίου του 2010, ενώ τον Φεβρουάριο του 2011 επανεκδόθηκε σε Android 2.3.3. Στην έκδοση αυτή υπάρχουν αλλαγές στο User Interface το οποίο έχει γίνει πιο απλό και ταχύ, ενώ υποστηρίζονται πλέον οθόνες μεγάλων μεγεθών και αναλύσεων. Υπάρχει πλέον το πρωτόκολλο SIP για κλήσεις μέσω VoIP, υποστηρίζεται ο τύπος βίντεο WebM/VP8 και ο κωδικοποιητής AAC, έχει βελτιωθεί ο ήχος καθώς και οι λειτουργίες απεικόνισης για την ανάπτυξη παιχνιδιών. Υπάρχει η δυνατότητα για Copy-Paste σε όλο το σύστημα και όχι μόνο στην ίδια εφαρμογή. Υποστηρίζεται το NFC (Near Field Communication) και η ύπαρξη πολλαπλών καμερών. Επίσης, έχει βελτιωθεί η ενεργειακή υποστήριξη και έχει γίνει μετάβαση από το σύστημα αρχείων YAFFS στο ext4 στις νέες συσκευές.

[3]



Εικόνα 9: Λογότυπο Gingerbread(bytenow.net) [9]

1.5.6 Android 3.0 Honeycomb (API Level: 13)

Η έκδοση “Honeycomb” παρουσιάστηκε στις 9 Μαΐου του 2011, με την ιδιαιτερότητα ότι προοριζόταν αποκλειστικά για tablets. Οι αλλαγές που έγιναν στην έκδοση αυτή έχουν να κάνουν κυρίως με τη βελτίωση της υποστήριξης των tablets. Υπάρχει ένα νέο, εντελώς διαφορετικό, User Interface και υποστηρίζονται διπύρρηνοι και τετραπύρρηνοι επεξεργαστές. Ακόμα, έχει απλοποιηθεί το multitasking έτσι ώστε ο χρήστης να μπορεί με τη χρήση ενός πλήκτρου (recent apps) να περνάει από μια εφαρμογή σε άλλη. Υπάρχει η δυνατότητα για Video Chat μέσω της εφαρμογής Google Talk καθώς η ανάγνωση βιβλίων μέσω του Google eBooks. Επιπλέον, μπορούν να κρυπτογραφηθούν όλα τα δεδομένα χρήστη.

[3]



Εικόνα 10: Λογότυπο Honeycomb(phandroid.com) [10]

1.5.7 Android 4.0 Ice cream sandwich (API Level: 15)

Η έκδοση “Ice Cream Sandwich” παρουσιάστηκε στις 19 Οκτωβρίου του 2011. Για άλλη μια φορά έχει βελτιωθεί η ταχύτητα και η απόδοση του συστήματος. Πλέον στο User Interface, το οποίο είναι και πάλι διαφορετικό, υπάρχουν εικονικά πλήκτρα τα οποία παίρνουν τη θέση των φυσικών ή αφής που υπήρχαν στις συσκευές. Βελτίωση της ασφάλειας του συστήματος με την προσθήκη αναγνώρισης προσώπου για να ξεκλειδώσει η συσκευή. Ο browser μπορεί να ανοίξει ταυτόχρονα μέχρι και 16 καρτέλες. Υπάρχει η δυνατότητα ο χρήστης να τερματίσει εφαρμογές οι οποίες τρέχουν στο background, ενώ μπορεί να θέσει και όρια στην κίνηση πακέτων δεδομένων. Η εφαρμογή Android Beam αξιοποιεί πλέον το NFC αφού επιτρέπει την αποστολή δεδομένων από τη συσκευή σε όσες βρίσκονται εντός μιας μικρής ακτίνας εμβέλειας. Ακόμα με την ύπαρξη του Wi-Fi Direct συσκευές μπορούν να συνδεθούν μεταξύ τους ασύρματα χωρίς την μεσολάβηση κάποιου access point. Τέλος, υποστηρίζεται η εγγραφή βίντεο σε 1080p. [3]



Εικόνα 11: Λογότυπο Ice cream sandwich(theinquirer.net) [11]

1.5.8 Android 4.1 Jelly Bean (API Level: 16)

Η έκδοση “Jelly Bean” παρουσιάστηκε στις 9 Ιουλίου 2012. Δύο ακόμα επανεκδόσεις έγιναν το Νοέμβριο του 2012 και τον Ιούλιο του 2013. Με έμφαση στη βελτίωση της απόδοσης σχεδιάστηκε για να δώσει στο λειτουργικό σύστημα μια ομαλότερη και καλύτερη ανταπόκριση. Υπάρχουν

βελτιώσεις στο σύστημα κοινοποίησης (notifications) που επιτρέπουν ειδοποιήσεις με κουμπιά δράσης. Επίσης βελτιώθηκαν τα γραφικά παιχνιδιών λόγω υποστήριξης της OpenGL ES 3.0 και η φωνητική αναζήτηση. Επίσης υπάρχει η δυνατότητα χρήσης της υπηρεσίας Google Wallet καθώς και η χρήση εξωτερικής συσκευής USB ήχου. [3][4]



Εικόνα 12: Λογότυπο Jelly Bean(gadgetfreak.gr) [12]

1.5.9 Android 4.4 Kit Kat (API Level: 19)

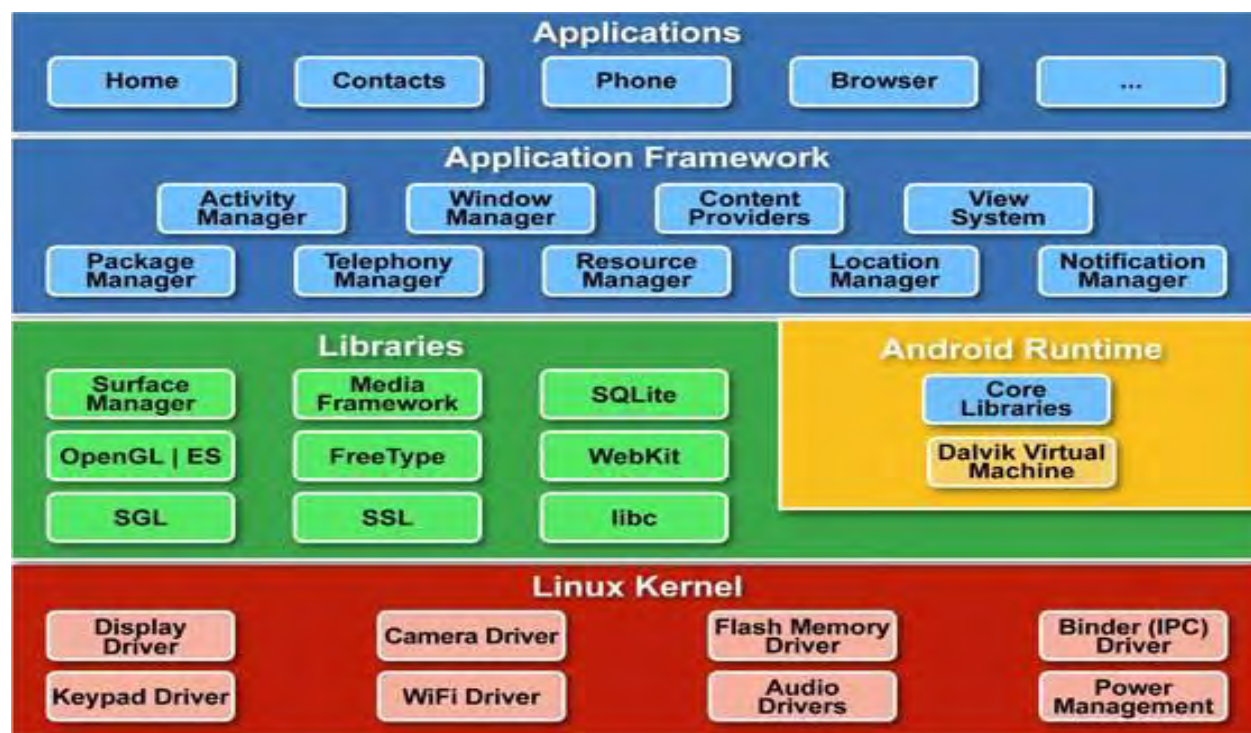
Η έκδοση "Kit Kat " παρουσιάστηκε στις 31 Οκτωβρίου 2013. Η συγκεκριμένη έκδοση διαθέτει μία εντελώς ανανεωμένη Phone app, που κάνει τις κλήσεις πιο εύκολες από ποτέ, βοηθώντας να γίνονται γρήγορα οι αναζητήσεις στις επαφές. Ενημερωμένη είναι και η εφαρμογή των Hangouts, που πλέον διαχειρίζεται όλα τα SMS και MMS. Βελτιώσεις υπάρχουν και στην υπηρεσία φωνητικών αναζητήσεων της Google. Οι αναζητήσεις στο νέο λειτουργικό είναι καλύτερες από ποτέ και πλέον μπορούν να οδηγούν ακόμη και σε εφαρμογές και όχι μόνο σε web pages. Το Google Now έχει σημαντικές βελτιώσεις στα χαρακτηριστικά των καρτελών του. Ανάμεσά τους ξεχωρίζουν καλύτερη ενσωμάτωση των υπηρεσιών cloud μέσα στις εφαρμογές (όπως π.χ. στην gallery της συσκευής), εμπλουτισμένη υποστήριξη printing, δυνατότητα να κάνει κάθε εφαρμογή να ενεργεί σαν μία NFC smart card, ενσωμάτωση υποστήριξης για αισθητήρες μέτρησης βημάτων, ένα νέο API για υποστήριξη IR, καθώς και ένα νέο utility screen recording. [3][4]



Εικόνα 13: Λογότυπο Kit Kat(droid-life.com) [13]

2 Αρχιτεκτονική Android

Το λειτουργικό σύστημα Android είναι μια στοίβα λογισμικού που περιλαμβάνει εφαρμογές, λειτουργικό σύστημα, run-time(χρόνος εκτέλεσης) περιβάλλον, ενδιάμεσο λογισμικό, υπηρεσίες και βιβλιοθήκες.[7]



Εικόνα 14: Η αρχιτεκτονική του Android(webgeometrics.com) [14]

Όπως βλέπουμε η αρχιτεκτονική του λειτουργικού συστήματος αποτελείται από 6 τμήματα.

- Τον πυρήνα Linux (Linux Kernel)
- Τις βιβλιοθήκες (Libraries)
- Την εικονική μηχανή Dalvik (Dalvik VM)
- Τον χρόνο εκτέλεσης (Android Runtime)
- Το πλαίσιο εφαρμογής (Application Framework)
- Τις εφαρμογές (Applications)

2.1 Πυρήνας Linux (Linux Kernel)

Στο κάτω μέρος της στοίβας είναι ο πυρήνας Linux. Δεν αλληλεπιδρά ποτέ με τους χρήστες και τους προγραμματιστές αλλά βρίσκεται στην καρδιά του συστήματος. Η σημασία του έγκειται στο γεγονός ότι παρέχει τις ακόλουθες λειτουργίες στο σύστημα Android:

- Αφαίρεση υλικού (Hardware Abstraction)
- Διαχείριση Μνήμης Προγραμμάτων
- Ρυθμίσεις ασφαλείας
- Λογισμικό Διαχείρισης Ισχύος
- Άλλους Οδηγούς υλικού
- Υποστήριξη για τις κοινές βιβλιοθήκες
- Δίκτυο στοίβας [7]

Android Version	Linux Kernel Version
1.0	2.6.25
1.5 (Cupcake)	2.6.27
1.6 (Donut)	2.6.29
2.2 (Froyo)	2.6.32
2.3 (Gingerbread)	2.6.35
3.0 (Honeycomb)	2.6.36
4.0.x (Ice Cream Sandwich)	3.0.1
4.1./4.2 (Jelly Bean)	3.0.31

Εικόνα 15: Διαφορετικές εκδόσεις Kernel(edureka.co) [15]

2.2 Βιβλιοθήκες

Το επόμενο στρώμα είναι οι βιβλιοθήκες του Android. Είναι αυτό το στρώμα που επιτρέπει στη συσκευή να χειριστεί διαφορετικούς τύπους δεδομένων. Αυτές οι βιβλιοθήκες είναι γραμμένες σε C ή C ++ και είναι ειδικά υλοποιημένες για ένα συγκεκριμένο υλικό.

Μερικές από τις σημαντικότερες βιβλιοθήκες περιλαμβάνουν τα εξής:

- Surface Manager: διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει απρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές.

- Media framework: προβλέπει διαφορετικά πολυμέσα κωδικοποιητών που επιτρέπουν την καταγραφή και αναπαραγωγή διαφόρων μορφών μέσων ενημέρωσης.
- SQLite: είναι ο μηχανισμός διαχείρισης βάσεων δεδομένων που χρησιμοποιούνται στο Android για σκοπούς αποθήκευσης δεδομένων.
- WebKit: Είναι η μηχανή του προγράμματος περιήγησης που χρησιμοποιείται για την εμφάνιση περιεχομένου HTML.
- OpenGL: χρησιμοποιείται για να καταστήσει 2D ή 3D γραφικά στην οθόνη.
- SSL: χρησιμοποιείται για να παρέχει ασφάλεια στο διαδίκτυο. [7]

2.3 Η εικονική μηχανή Dalvik

Το Dalvik είναι λογισμικό ανοιχτού κώδικα που πήρε το όνομά του από το ψαροχώρι της Dalvík στο Eyjafjörður της Ισλανδίας. Είναι το λογισμικό που ευθύνεται για τη λειτουργία των εφαρμογών για τις Android συσκευές.

- πρόκειται για ένα μητρώο που βασίζεται σε Virtual Machine
- είναι βελτιστοποιημένο για χαμηλές απαιτήσεις μνήμης
- έχει σχεδιαστεί για να επιτρέπει πολλαπλές εμφανίσεις VM σε ταυτόχρονη εκτέλεση
- στηρίζεται σε λειτουργικό σύστημα για την απομόνωση της διαδικασίας, διαχείριση μνήμης και υποστήριξη νημάτων
- λειτουργεί με αρχεία DEX[6]

2.4 Χρόνος Εκτέλεσης Εφαρμογής (Android Runtime)

Ο χρόνος εκτέλεσης των εφαρμογών του Android, βρίσκεται στο ίδιο επίπεδο με τις κύριες βιβλιοθήκες και την μηχανή Dalvik. Εδώ βρίσκουμε το κοινό σημείο επαφής μεταξύ των δυνατοτήτων που παρέχουν οι βιβλιοθήκες και του χρόνου εκτέλεσης της εικονικής μηχανής Dalvik. [7]

2.5 Πλαίσιο Εφαρμογής (Application Framework)

Οι εφαρμογές μας αλληλεπιδρούν άμεσα με τα παραπάνω τμήματα της αρχιτεκτονικής του Android. Τα προγράμματα αυτά διαχειρίζονται τις βασικές λειτουργίες του τηλεφώνου όπως η διαχείριση πόρων, η διαχείριση φωνητικών κλήσεων κ.λπ.

Σημαντικά τμήματα του πλαισίου εφαρμογής:

- Activity Manager (Διαχειριστής Δραστηριοτήτων): διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και παρέχει δυνατότητα πλοήγησης από δραστηριότητα σε δραστηριότητα κρατώντας αποθηκευμένη στη μνήμη τη σειρά εκτέλεσης αυτών
- Content Provider(Πάροχος Περιεχομένου): δίνει τη δυνατότητα στις εφαρμογές να μοιράζονται ή να ανταλλάσσουν δεδομένα μιας συγκεκριμένης μορφής η οποία ορίζεται από τον πάροχο
- Telephony Manager(Διαχειριστής Τηλεφώνου): διαχειρίζεται όλες τις φωνητικές κλήσεις. Τον χρησιμοποιούμε αν θέλουμε να έχουν πρόσβαση στη εφαρμογή μας οι φωνητικές κλήσεις
- Location Manager(Διαχειριστής Τοποθεσίας): διαχειρίζεται την τοποθεσία χρησιμοποιώντας GPS ή cell tower
- Resource Manager(Διαχειριστής Πόρων): παρέχει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα όπως πχ, εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ [7]

2.6 Εφαρμογές (Applications)

Οι εφαρμογές βρίσκονται στο ανώτερο στρώμα της στοίβας. Ένας μέσος χρήστης της Android συσκευής αλληλεπιδρά κυρίως με αυτό το στρώμα (για τις βασικές λειτουργίες, όπως η πραγματοποίηση τηλεφωνικών κλήσεων, πρόσβαση στο πρόγραμμα περιήγησης στο Web, κλπ). Τα πιο κάτω στρώματα είναι προσβάσιμα ως επί το πλείστον από τους προγραμματιστές. [7]

Πολλές εφαρμογές έρχονται εγκαταστημένες με κάθε συσκευή, όπως:

- SMS client app
- Dialer
- Web browser
- Contact manager

3 Εργαλεία ανάπτυξης της εφαρμογής

Η γλώσσα προγραμματισμού που χρησιμοποιείται για την κατασκευή εφαρμογών Android είναι η Java. Οπότε βασική προϋπόθεση για την ανάπτυξη μίας εφαρμογής είναι η διαθεσιμότητα των αντίστοιχων εργαλείων της γλώσσας προγραμματισμού που θα χρησιμοποιηθούν και συγκεκριμένα το Java Development Kit (JDK).

Επίσης απαιτείται ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για να μεταγλωττίζει και να εκτελεί τα προγράμματα. Στη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε το Eclipse.

Βασικό εργαλείο αποτελεί το Android software development kit (SDK) το οποίο παρέχει τα επιπλέον εργαλεία που χρειαζόμαστε για την ανάπτυξη εφαρμογής Android.. Η σύνδεση του Android SDK με το γραφικό περιβάλλον γίνεται μέσω της επέκτασης Android Development Tools ή ADT Plugin που εγκαθιστάμε στο Eclipse.

Η ADT Plugin μας δίνει τη δυνατότητα να εγκαταστήσουμε τα εργαλεία για την υλοποίηση της εφαρμογής σε οποιαδήποτε έκδοση Android.

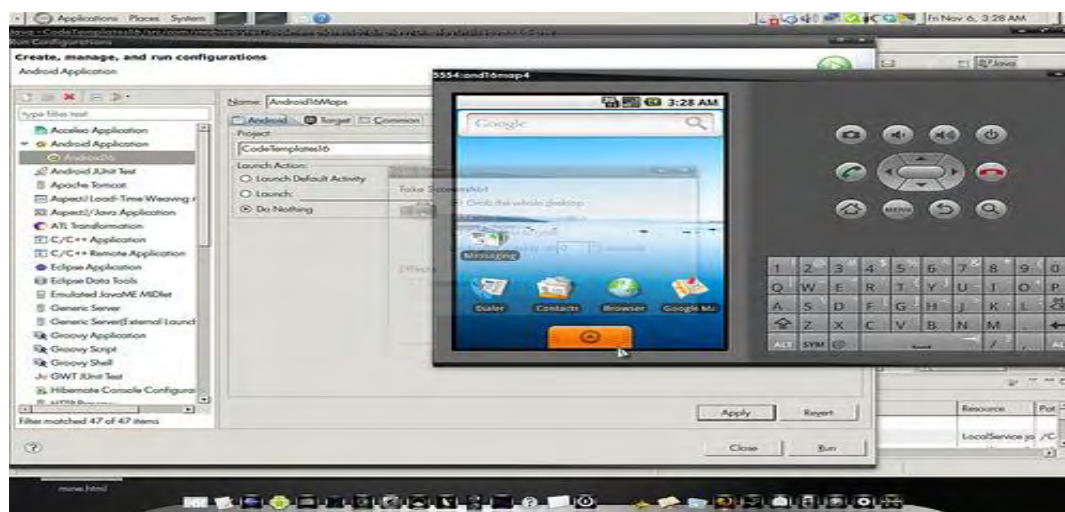
Τέλος, για να δοκιμάσουμε την εφαρμογή και να δούμε τα αποτελέσματα της την εγκαθιστάμε και την εκτελούμε κατ' ευθείαν σε φυσική συσκευή όπως κινητό τηλέφωνο ή tablet που χρησιμοποιεί το λειτουργικό σύστημα Android. Φυσικά υπάρχει και η δυνατότητα να δημιουργήσουμε μια εικονική συσκευή Android (Android Virtual Device ή AVD) η οποία ουσιαστικά αποτελεί προσομοιωτή τόσο software όσο και hardware ενός κινητού τηλεφώνου με λειτουργικό σύστημα Android. Την συσκευή αυτή, την εγκαθιστάμε μέσα από την ADT Plugin, από όπου μπορούμε να ρυθμίσουμε πολλές παραμέτρους της, όπως την έκδοση του Android που θα χρησιμοποιεί, το μέγεθος της οθόνης, το μέγεθος της κάρτας SD και της cache κ.α.

3.1 Android SDK

Το kit (SDK) του λογισμικού Android περιλαμβάνει ένα ολοκληρωμένο σύνολο εργαλείων ανάπτυξης. Αυτά περιλαμβάνουν ένα πρόγραμμα εντοπισμού σφαλμάτων, βιβλιοθήκες, έναν εξομοιωτή συσκευής, documentation, δείγματα κώδικα και tutorials.

Περιέχει παραδείγματα εφαρμογών με τον πηγαίο τους κώδικα, εργαλεία ανάπτυξης, ένα εξομοιωτή και τις απαιτούμενες βιβλιοθήκες για την ανάπτυξη των εφαρμογών στη γλώσσα προγραμματισμού Java. Ακόμη αναλαμβάνει τη μεταγλώττιση του πηγαίου κώδικα ώστε να τρέχει στην εικονική μηχανή Dalvik. Το Android SDK περιλαμβάνει μια κινητή συσκευή emulator - μια εικονική φορητή συσκευή που τρέχει στον υπολογιστή σας. Ο εξομοιωτής επιτρέπει να αναπτυχθούν και να δοκιμαστούν Android εφαρμογές χωρίς τη χρήση μιας φυσικής συσκευής. Ο εξομοιωτής του Android χρησιμοποιεί τα Android Virtual Device (AVD). Τα AVD επιτρέπουν τον ορισμό διαφόρων πτυχών του υλικού του μιμούμενου τηλεφώνου και τη δημιουργία πολλών συνθέσεων.

Όταν η εφαρμογή λειτουργεί με τον εξομοιωτή, μπορεί να χρησιμοποιήσει τις υπηρεσίες της πλατφόρμας Android για να επικαλεστεί άλλες εφαρμογές, να έχει πρόσβαση στο δίκτυο, να αναπαράγει ήχο και βίντεο, να αποθηκεύει και να ανακτά δεδομένα, να ενημερώνει το χρήστη και άλλα. Ο εξομοιωτής περιλαμβάνει επίσης μια ποικιλία από δυνατότητες εντοπισμού σφαλμάτων, όπως μια κονσόλα στην οποία καταγράφονται τα μηνύματα εξόδου του πυρήνα, προσομοίωση τυχόν διακοπών της εφαρμογής (όπως όταν φθάνουν μηνύματα SMS ή τηλεφωνικές κλήσεις), καθώς και απορρίψεις στο δίκτυο δεδομένων.[9][11]



Εικόνα 16: Eclipse and Android SDK(flickr.com)[16]

3.2 Android NDK

Το NDK είναι ένα σύνολο εργαλείων που επιτρέπει να εφαρμόσουμε μέρη της εφαρμογής χρησιμοποιώντας *native* γλώσσες προγραμματισμού όπως η C και C++. Για ορισμένους τύπους εφαρμογών, αυτό μπορεί να είναι χρήσιμο, γιατί επιτρέπει την επαναχρησιμοποίηση βιβλιοθηκών. Πάραυτα οι περισσότερες εφαρμογές δεν χρειάζονται το Android NDK.

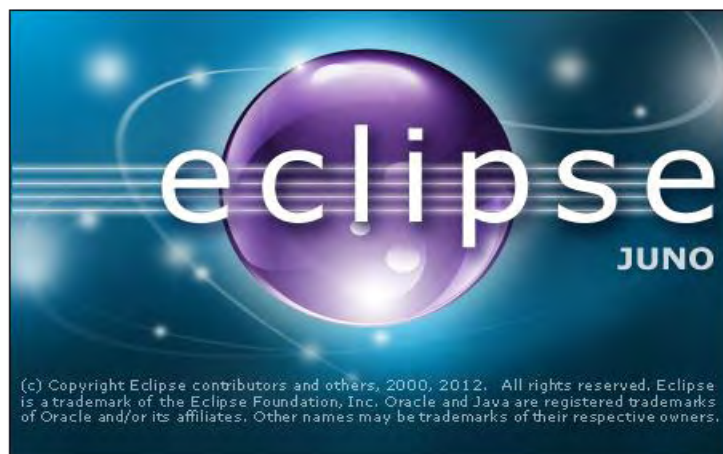
Θα πρέπει να κατανοήσουμε ότι το NDK δεν θα ωφελήσει τις περισσότερες εφαρμογές. Για καλύτερο αποτέλεσμα ένας προγραμματιστής θα πρέπει να εξισορροπήσει τα οφέλη του και τα μειονεκτήματά του. Αξίζει να σημειωθεί ότι χρησιμοποιώντας το *native* κώδικα για το Android δεν οδηγούμαστε σε αισθητή βελτίωση της απόδοσης, αλλά αυξάνεται πάντα η πολυπλοκότητα της εφαρμογής. Σε γενικές γραμμές, θα πρέπει να χρησιμοποιούμε το NDK μόνο εάν είναι απαραίτητο για την εφαρμογή.

Τυπικά καλές υποψήφιες εφαρμογές για το NDK είναι οι εφαρμογές όπως μηχανές παιχνιδιών, επεξεργασίας σήματος καθώς επίσης και η προσομοίωση της φυσικής οι οποίες είναι αυτόνομες και χρησιμοποιούν λίγη μνήμη.[9][12]

3.3 Eclipse

Το Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment). Περιέχει μια βάση εργασίας και ένα επεκτάσιμο σύστημα *plug-in* για την προσαρμογή του περιβάλλοντος. Χρησιμοποιείται κυρίως για προγραμματισμό σε Java και για την ανάπτυξη εφαρμογών. Με τη βοήθεια των διαφόρων *plug-ins*, το Eclipse μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε άλλες γλώσσες προγραμματισμού όπως Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, Natural, Perl, PHP, Prolog, Python, R, Ruby, Scala, Clojure, Groovy, Scheme και Erlang.

Εκδίδεται σύμφωνα με τους όρους του Eclipse Public License, το Eclipse SDK είναι ελεύθερο και ανοικτού κώδικα λογισμικό. Ήταν ένα από τα πρώτα IDEs που τρέχει κάτω από GNU Classpath, επίσης τρέχει χωρίς προβλήματα κάτω από IcedTea.[10]



Εικόνα 17: Eclipse IDE(javapapo.blogspot.gr)[17]

3.4 Android Development Tools (ADT)

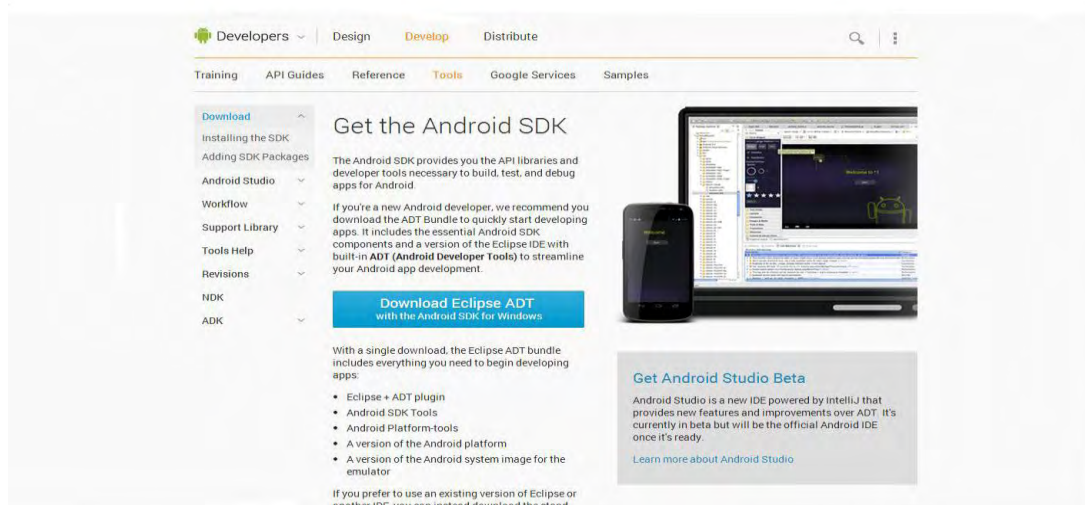
Η Google υποστηρίζει επίσημα το Eclipse και έχει αναπτύξει ειδικά για αυτό ένα επιπρόσθετο άρθρωμα, το Android Development Tools, το οποίο συνδέει το Eclipse με το Android SDK και όλες τις δυνατότητες του.

Το ADT (Android Developer Tools) είναι ένα plugin για το Eclipse που παρέχει μια σειρά από εργαλεία που ενσωματώνονται με το Eclipse IDE. Προσφέρει πρόσβαση σε πολλές λειτουργίες που βοηθούν να αναπτύξουμε εφαρμογές Android. Το ADT παρέχει πρόσβαση GUI σε πολλά από τα εργαλεία της γραμμής εντολών SDK, καθώς και ένα εργαλείο σχεδιασμού UI για την ταχεία κατασκευή πρωτοτύπων, το σχεδιασμό και την οικοδόμηση της διεπαφής χρήστη των εφαρμογών.[13]

3.5 Εγκατάσταση του Eclipse IDE

Για την εγκατάσταση του Eclipse IDE μεταβαίνουμε στην επίσημη ιστοσελίδα της Google στο σύνδεσμο <http://developer.android.com/sdk/installing/installing-adt.html>. Επίσης θα πρέπει να είναι εγκαταστημένο και το jdk και το JRE , τα οποία τα

κατεβάζουμε από την ιστοσελίδα της Oracle
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.



Εικόνα 18: Eclipse download

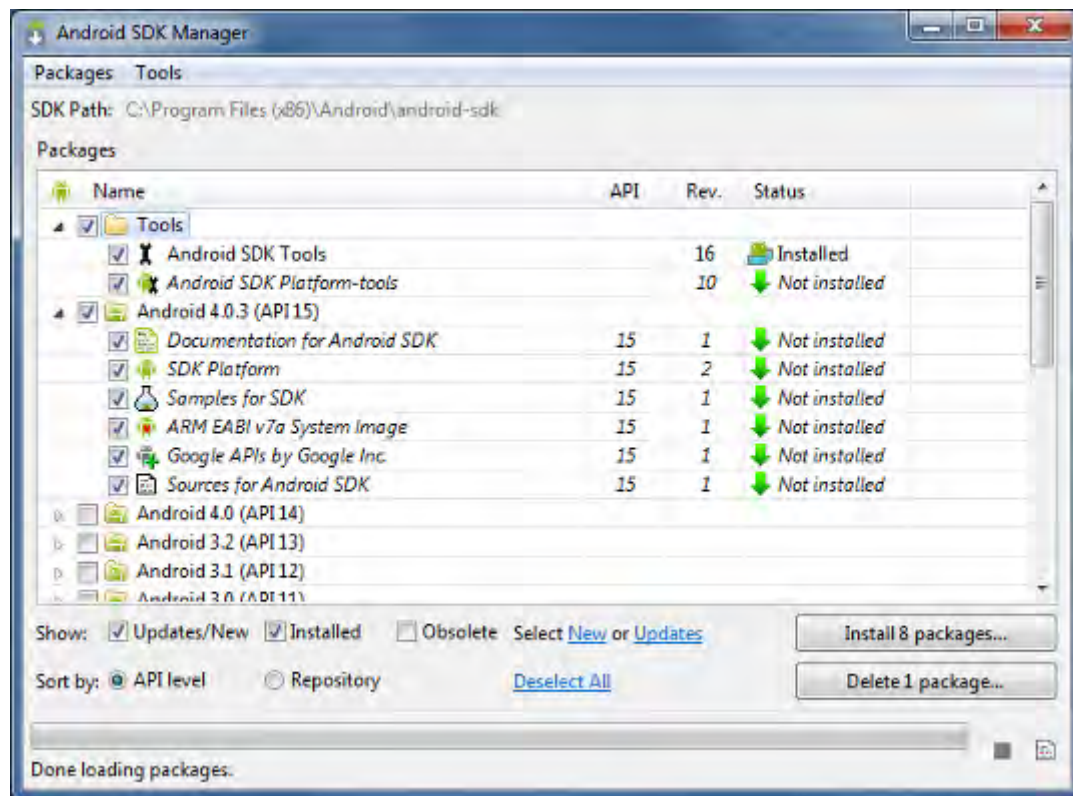


Εικόνα 19: JDK download(courses.cs.washington.edu)[18]

3.6 Εγκατάσταση του Android SDK Manager

Το Android SDK διαχωρίζει τα εργαλεία, τις πλατφόρμες, και άλλα σε πακέτα που μπορούμε να κατεβάσουμε χρησιμοποιώντας τον SDK Manager. Για παράδειγμα, όταν τα εργαλεία SDK είναι ενημερωμένα ή μια νέα έκδοση της πλατφόρμας Android έχει κυκλοφορήσει, μπορούμε να χρησιμοποιήσουμε τον SDK Manager για να τα κατεβάσουμε γρήγορα στο περιβάλλον.

Μπορούμε να εκκινήσουμε τον SDK Manager από το Eclipse (με ADT) επιλέγοντας **Window > Android SDK Manager**.

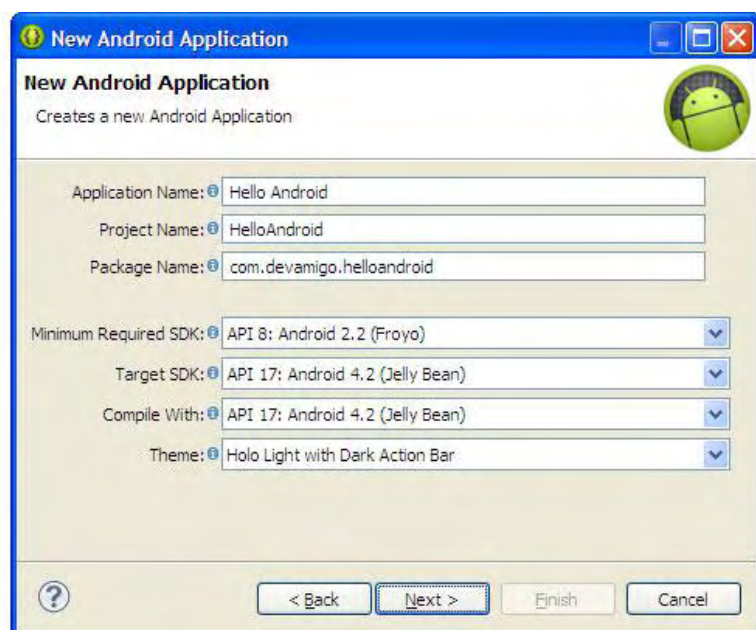
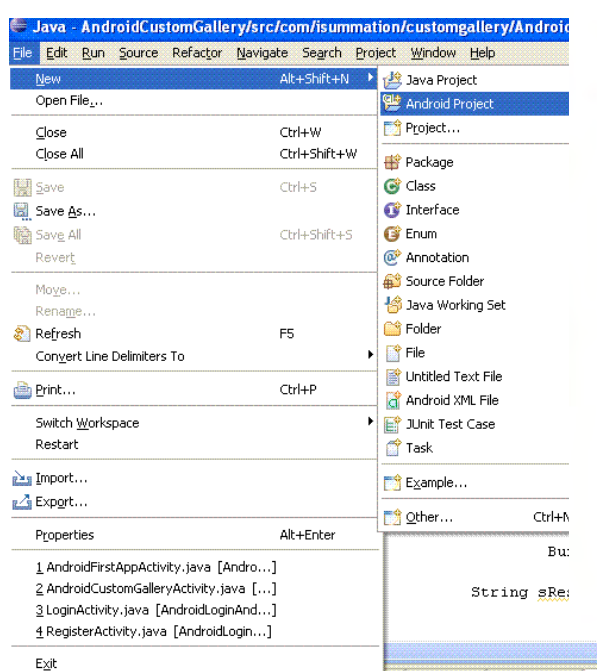


Εικόνα 20: Android SDK Manager(developer.android.com)[19]

3.7 Δημιουργία νέου Project

Για να ξεκινήσουμε ένα καινούργιο Project στο Eclipse επιλέγουμε **File >New>Android Application Project**.

Στο παράθυρο που ανοίγει βάζουμε το όνομα της εφαρμογής του Project και του πακέτου. Στην συνέχεια, επιλέγουμε την μικρότερη και μεγαλύτερη έκδοση στην οποία θέλουμε να τρέχει η εφαρμογή μας και πατάμε **Next**. Επιλέγουμε την εικόνα της εφαρμογής και πατάμε **Next**. Διαλέγουμε **Blank activity** και πατάμε **Next**. Επιλέγουμε τα ονόματα της Activity και του Layout και πατάμε **Finish**.



Εικόνα 21: Δημιουργία νέου Project(javasrilankansupport.com)[20]

4 Δομή της εφαρμογής Android Locator

4.1 Η κλάση AsyncTask

Η AsyncTask είναι μια αφηρημένη κλάση που παρέχεται από το Android η οποία βοηθά να χρησιμοποιούμε το UI thread σωστά. Η κλάση αυτή επιτρέπει να εκτελούμε μεγάλες λειτουργίες και δείχνει τα αποτελέσματά της στο UI thread χωρίς να χρειάζεται να χειριστεί τα νήματα.

AsyncTask έχει σχεδιαστεί για να είναι μια κλάση αρωγός γύρω από το Thread και τον Handler και δεν αποτελεί ένα γενικό πλαίσιο threading. Η AsyncTask πρέπει ιδανικά να χρησιμοποιείται για μικρής διάρκειας χειρισμούς (λίγα δευτερόλεπτα το πολύ.) Αν θέλουμε να κρατήσουμε τα θέματα που τρέχουν για μεγάλο χρονικό διάστημα, συνιστάται ιδιαίτερα να χρησιμοποιήσουμε τα διάφορα APIs που παρέχονται από το πακέτο `java.util.concurrent` όπως `ThreadPoolExecutor` και `FutureTask`.

Κάθε φορά που μια εφαρμογή Android ξεκινάει ένα νήμα δημιουργείται. Υποθέτοντας ότι κάνουμε τη λειτουργία του δικτύου με ένα κλικ σε ένα κουμπί στην εφαρμογή. Με το κλικ στο κουμπί μια αίτηση θα πρέπει να γίνει στο διακομιστή και η απάντηση θα αναμένεται. Λόγω του ενιαίου μοντέλου νήματος του Android, μέχρι το χρόνο απόκρισης αναμένεται η οθόνη μας να μην αποκρίνεται. Γι' αυτό και θα πρέπει να αποφεύγεται η εκτέλεση πολύχρονων εργασιών στο UI thread. Αυτό περιλαμβάνει το αρχείο και την πρόσβαση στο δίκτυο.

Για να ξεπεραστεί αυτό μπορούμε να δημιουργήσουμε νέο νήμα και να εφαρμόσουμε τη μέθοδο εκτέλεσης για να εκτελέσουμε αυτήν την κλήση του δικτύου, έτσι το UI θα εξακολουθήσει να ανταποκρίνεται. Αλλά δεδομένου ότι το Android ακολουθεί το ενιαίο μοντέλο νήματος και το Android UI Toolkit δεν είναι ασφαλή νήμα, οπότε αν υπάρχει ανάγκη να κάνουμε κάποια αλλαγή στην UI με βάση το αποτέλεσμα της λειτουργίας που εκτελείται, τότε αυτή η προσέγγιση μπορεί να οδηγήσει σε κάποια θέματα.

Η AsyncTask έχει τέσσερα στάδια:

1. `onPreExecute`: Αυτή η μέθοδος καλείται πριν την μέθοδο `doInBackground`.
2. `doInBackground`: Η `doInBackground` χρησιμοποιείται για υπολογισμούς που παίρνουν ώρα να εκτελεστούν, δηλαδή είναι το βασικό βήμα για να εκτελέσουμε τον κώδικα που θέλουμε. Οι παράμετροι της AsyncTask

παίρνονται από αυτό το στάδιο και το αποτέλεσμα επιστρέφεται στο επόμενο βήμα που αναλύουμε παρακάτω.

3. `onPostExecute`: Αυτή η μέθοδος καλείται μετά την ολοκλήρωση της κλήσης της `doInBackground`. Το αποτέλεσμα από την `doInBackground` περνιέται σε αυτή τη μέθοδο.
4. `onProgressUpdate`: Αυτή η μέθοδος καλείται καλώντας την `publishProgress` ανά πάσα στιγμή ενώ εκτελείται η `doInBackground`. Συνήθως η `onProgressUpdate` χρησιμοποιείται για να ενημερώσουμε το UI Thread με κάποια μπάρα προόδου ενώ εκτελείται η `doInBackground`. [18][19]

4.2 Android Manifest

Κάθε εφαρμογή πρέπει να έχει ένα αρχείο `AndroidManifest.xml` (με ακριβώς αυτό το όνομα) στον αρχικό του κατάλογο. Το αρχείο `Manifest` παρουσιάζει ουσιώδεις πληροφορίες σχετικά με την εφαρμογή στο σύστημα Android, πληροφορίες που πρέπει να έχει το σύστημα για να μπορέσει να τρέξει οποιοδήποτε κώδικα της εφαρμογής. Μεταξύ άλλων, το `Manifest` κάνει τα εξής:

- Ονομάζει το πακέτο της Java για την εφαρμογή. Το όνομα του πακέτου χρησιμεύει ως μοναδικό αναγνωριστικό για την εφαρμογή.
- Περιγράφει τις συνιστώσες της εφαρμογής - activities, services, broadcast receivers, and content providers από τις οποίες αποτελείται η εφαρμογή. Ονομάζει τις κλάσεις που υλοποιούν κάθε μια από τις συνιστώσες και δημοσιεύει τις ικανότητές τους (για παράδειγμα, ποια Intent messages μπορούν να χειριστούν). Οι δηλώσεις αυτές επιτρέπουν στο σύστημα Android να γνωρίζει ποιες είναι οι συνιστώσες και υπό ποιες προϋποθέσεις μπορούν να ξεκινήσουν.
- Καθορίζει ποιες διεργασίες θα φιλοξενήσουν τις συνιστώσες της εφαρμογής.
- Δηλώνει τα δικαιώματα που πρέπει να έχει η εφαρμογή ώστε να έχει πρόσβαση σε προστατευόμενα μέρη του API και να αλληλεπιδρά με άλλες εφαρμογές.
- Επίσης δηλώνει τα δικαιώματα που οι άλλοι πρέπει να έχουν για να αλληλεπιδρούν με τα στοιχεία της εφαρμογής.
- Παραθέτει τις κλάσεις που παρέχουν χαρακτηριστικά του προφίλ και άλλες πληροφορίες ενώ η εφαρμογή εκτελείται. Αυτές οι δηλώσεις είναι παρούσες στο `Manifest` μόνο, ενώ η εφαρμογή έχει αναπτυχθεί και έχει δοκιμαστεί.

- Δηλώνει το ελάχιστο επίπεδο του Android API που απαιτεί η εφαρμογή.
- Παραθέτει τις βιβλιοθήκες που η εφαρμογή θα πρέπει να συνδεστεί.[16][17]

Το αρχείο AndroidManifest.xml της εφαρμογής:

```

xx<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="info.androidhive.googlemapsv2"
    android:versionCode="1"
    android:versionName="1.0" >

    <permission
        android:name="info.androidhive.googlemapsv2.permission.MAPS_RECEIVE"
        android:protectionLevel="signature" />

    <uses-permission
        android:name="info.androidhive.googlemapsv2.permission.MAPS_RECEIVE" />

    <uses-sdk
        android:minSdkVersion="11"
        android:targetSdkVersion="20" />

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
    />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <!-- Required to show current location -->
    <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
    />
    <!-- Required OpenGL ES 2.0. for Maps V2 -->
    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />

    <!-- Requires OpenGL ES version 2 -->
    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="info.androidhive.googlemapsv2.MainActivity"
            android:label="@string/app_name" >

    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

```

```

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<provider
    android:name="LocationsContentProvider"
    android:authorities="info.androidhive.googlemapsv2.locations"
    android:exported="false" />

<!-- Google API Key -->
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyCgzSMgDJDmUfTLu1hzulP5IkYWDeraEA" />
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
</application>
</manifest>

```

4.3 Activity

Activity είναι ένα συστατικό της εφαρμογής που παρέχει μια οθόνη με την οποία οι χρήστες μπορούν να αλληλεπιδρούν με σκοπό να εκτελεστεί μία πράξη, όπως να καλέσουμε στο τηλέφωνο, να τραβήξουμε μια φωτογραφία, να στείλουμε ένα e-mail, ή να δούμε ένα χάρτη. Σε κάθε activity υπάρχει ένα παράθυρο. Το παράθυρο συνήθως γεμίζει την οθόνη, αλλά μπορεί να είναι μικρότερο από την οθόνη και να είναι πάνω από άλλα παράθυρα.

Μια εφαρμογή αποτελείται συνήθως από πολλαπλές activities που είναι χαλαρά συνδεδεμένες ή μία με την άλλη. Συνήθως, μια activity σε μια εφαρμογή ορίζεται ως η «κύρια» activity, η οποία παρουσιάζεται στο χρήστη κατά την εκκίνηση της εφαρμογής για πρώτη φορά. Κάθε activity μπορεί στη συνέχεια να αρχίσει μια άλλη activity, προκειμένου να εκτελέσει διάφορες ενέργειες. Κάθε φορά που ξεκινά μια νέα activity η προηγούμενη έχει σταματήσει αλλά το σύστημα διατηρεί την activity σε μια στοίβα (το "back stack"). Όταν ξεκινά μια νέα activity ωθείται στο πίσω μέρος της στοίβα και παίρνει την προσοχή του χρήστη. Το πίσω μέρος στοίβας τηρεί το βασικό μηχανισμό στοίβας "last in, first out", έτσι ώστε όταν ο χρήστης τελειώσει με την τρέχουσα activity και πιάσει το κουμπί πίσω, αυτή έχει πεταχτεί από τη στοίβα και η προηγούμενη activity επανέρχεται.

Όταν μια activity έχει διακοπεί επειδή μια νέα activity ξεκινά, έχει ειδοποιηθεί από την αλλαγή αυτή μέσω των μεθόδων του κύκλου ζωής επανάκλησης της activity αυτής. Υπάρχουν διάφορες μέθοδοι επανάκλησης που θα μπορούσαν να λάβουν μια activity, λόγω αλλαγής της κατάστασης της-κατά πόσον το σύστημα αυτό δημιουργεί, σταματά, επαναξεκινά ή καταστρέφει. Για παράδειγμα, όταν σταματήσει η activity θα πρέπει να ελευθερωθούν μεγάλα αντικείμενα, όπως οι συνδέσεις του δικτύου ή μια βάση δεδομένων. Όταν η activity επανέλθει θα μπορεί να εξαγοράσει τους απαραίτητους πόρους και να συνεχίσει τις ενέργειες που είχαν διακοπεί. Αυτές οι μεταβάσεις είναι όλα μέρος του κύκλου ζωής της activity.[8][20][21]

4.3.1 Fragments

Ένα Fragment είναι ένα κομμάτι μιας activity που επιτρέπει την ξεχωριστή σχεδίαση της activity. Δεν είναι λάθος να πούμε ότι ένα fragment είναι ένα είδος sub-activity.

Τα επόμενα αποτελούν σημαντικά σημεία του fragment.

- Ένα fragment έχει τη δική του διαρρύθμιση και τη δική του συμπεριφορά.
- Μπορούμε να προσθέσουμε ή να αφαιρέσουμε fragments σε μια activity ενώ βρίσκεται σε λειτουργία.
- Μπορούμε να συνδυάσουμε πολλά fragments σε μια activity για να οικοδομήσουμε ένα multi-pane UI.
- Ένα fragment μπορεί να χρησιμοποιηθεί σε πολλαπλές activities.
- Ο κύκλος ζωής του fragment είναι στενά συνδεδεμένος με τον κύκλο ζωής της activity που σημαίνει ότι όταν η activity έχει διακοπεί, όλα τα fragments που είναι διαθέσιμα στην activity σταματούν.
- Ένα fragment μπορεί να υλοποιήσει μια συμπεριφορά που δεν έχει κανένα στοιχείο της διεπαφής χρήστη.
- Fragments προστέθηκαν στο Android στην Honeycomb έκδοση(API 11).[8][24]

4.4 Resources

Στα resources μιας εφαρμογής ορίζεται το layout των activities, οι διάφορες εικόνες και λεκτικά που χρησιμοποιούνται στα activities. Σε κάθε activity αντιστοιχεί ένα Layout αρχείο, το οποίο περιγράφει τη θέση των διάφορων αντικειμένων στην οθόνη. Το layout αρχείο είναι ένα αρχείο XML. Στην πράξη το αρχείο αυτό διαμορφώνεται από κατάλληλους γραφικούς editors που προσφέρονται από ολοκληρωμένα περιβάλλοντα ανάπτυξης όπως το Eclipse. Στην ενότητα για την διεπαφή χρήστη αναφέρθηκε ότι η διάταξη των γραφικών στοιχείων δηλώνεται σε αρχεία xml. Συγκεκριμένα στο project της εφαρμογής μας υπάρχει ο φάκελος res/layout/ στον οποίο τοποθετούμε όλα τα αρχεία xml που αφορούν στο user interface της εφαρμογής μας (το res προέρχεται από το resources). Η εντολή που χρησιμοποιείται για να δηλωθεί το αρχείο xml που θα χρησιμοποιηθεί σε μία activity είναι η setContentView().[8][23]

4.5 Services

Ένα service είναι ένα στοιχείο της εφαρμογής που μπορεί να εκτελέσει μακροχρόνιες εργασίες στο παρασκήνιο και δεν παρέχει ένα περιβάλλον εργασίας χρήστη. Ένα άλλο στοιχείο της εφαρμογής μπορεί να ξεκινήσει ένα service το οποίο θα συνεχίσει να τρέχει στο παρασκήνιο, ακόμη και αν ο χρήστης μεταβεί σε μια άλλη εφαρμογή. Επιπλέον, ένα συστατικό μπορεί να δεσμευτεί σε ένα service για να αλληλεπιδράσει με αυτό. Για παράδειγμα ένα service που θα μπορούσε να χειριστεί τις συναλλαγές του δικτύου, την αναπαραγωγή μουσικής, την εκτέλεση αρχείου I / O ή να αλληλεπιδράσει με έναν πάροχο περιεχομένου, όλα από το παρασκήνιο.

Ένα service μπορεί να λάβει δύο μορφές:

- Started

Ένα service ξεκινάει όταν ένα στοιχείο της εφαρμογής όπως μια activity. Το service ξεκινά με την κλήση της startService (). Μόλις ξεκινήσει, ένα service μπορεί να τρέχει στο παρασκήνιο επ 'αόριστον, ακόμη και αν το στοιχείο που το άρχισε καταστραφεί.

- Bound

Ένα service είναι "bound" όταν ένα στοιχείο της εφαρμογής δεσμεύεται καλώντας την bindService (). Ένα bound service λειτουργεί μόνο για όσο διάστημα ένα άλλο στοιχείο της εφαρμογής είναι συνδεδεμένο με αυτό. Πολλαπλά συστατικά μπορεί να δεσμευτούν σε ένα service την

φορά αλλά όταν όλα αυτά αποδεσμευτούν το service έχει καταστραφεί.[8][27]

4.6 Android SQLite Database

Για την αποθήκευση των τοποθεσιών σε μια βάση δεδομένων χρησιμοποιήθηκε η SQLite database. Περιέχει κλάσεις διαχείρισης δεδομένων που χρησιμοποιεί η εφαρμογή(Locator) για να διαχειριστεί την βάση της. Η SQLite είναι ενσωματωμένη σε κάθε συσκευή Android. Χρησιμοποιώντας μια βάση δεδομένων SQLite στο Android δεν απαιτείται καμία διαδικασία ρύθμισης ή διαχείρισης της βάσης δεδομένων. Το μόνο που πρέπει να καθοριστεί είναι οι δηλώσεις SQL για τη δημιουργία και την ενημέρωση της βάσης δεδομένων. Στη συνέχεια η βάση δεδομένων διαχειρίζεται αυτόματα για εμάς από την πλατφόρμα του Android. Η πρόσβαση σε μια βάση δεδομένων SQLite περιλαμβάνει την πρόσβαση στο σύστημα αρχείων. Αυτό μπορεί να είναι αργό. Ως εκ τούτου, συνιστάται να εκτελέσετε λειτουργίες της βάσης δεδομένων ασύγχρονα.[14][15]

4.7 Content providers

Οι content providers διαχειρίζονται την πρόσβαση σε ένα διαρθρωμένο σύνολο δεδομένων. Ενσωματώνουν τα δεδομένα και παρέχουν μηχανισμούς για τον προσδιορισμό της ασφάλειας των δεδομένων. Content providers είναι η τυπική διεπαφή που συνδέει τα δεδομένα σε μια διαδικασία με τον κώδικα που τρέχει σε μια άλλη διαδικασία.

Οι εφαρμογές μπορούν να αποθηκεύσουν τα δεδομένα τους σε αρχεία, στη βάση δεδομένων SQLite, σε preferences ή χρησιμοποιώντας οποιονδήποτε άλλο μηχανισμό που τους παρέχει αυτή τη δυνατότητα. Ένας content provider επίσης είναι χρήσιμος εάν θέλουμε τα δεδομένα μιας εφαρμογής να γίνουν διαθέσιμα και σε άλλες εφαρμογές. Ένας content provider είναι μια κλάση που υλοποιεί ένα standard set από μεθόδους, οι οποίες επιτρέπουν σε άλλες εφαρμογές να αποθηκεύουν και να ανακτούν τον τύπο δεδομένων που χειρίζεται ο content provider.[8]

4.8 Διεπαφή χρήστη

[8][22] Η διεπαφή χρήστη αποτελεί την τελική εικόνα που βλέπει ο χρήστης, το γραφικό περιβάλλον στο οποίο περιηγείται και ενεργοποιεί όλες τις λειτουργίες της εφαρμογής.

Το user interface και η λειτουργικότητα της εφαρμογής αποτελούν αλληλένδετα στοιχεία και χωρίς το ένα δε μπορεί να υπάρξει το άλλο. Πολλές φορές μάλιστα είναι δυσκολότερος ο σχεδιασμός ενός όμορφου και εύχρηστου περιβάλλοντος εργασίας, παρά η ίδια η λειτουργικότητα της εφαρμογής. Καθίσταται σαφές, λοιπόν, ότι απαιτεί μεγάλη προσπάθεια και προσοχή η δημιουργία ενός γραφικού περιβάλλοντος που θα προσελκύει τους χρήστες και θα τους ωθεί να χρησιμοποιούν μία συγκεκριμένη εφαρμογή έναντι μίας άλλης, με τις ίδιες λειτουργίες.

4.8.1 Layout

Πρωταρχικό στοιχείο του γραφικού περιβάλλοντος αποτελεί η διάταξη των γραφικών στοιχείων ή layout. Υπάρχουν 4 είδη layout, τα LinearLayout (γραμμική διάταξη), RelativeLayout (σχετική διάταξη), FrameLayout (διάταξη πλαισίου) και TableLayout (διάταξη πίνακα).

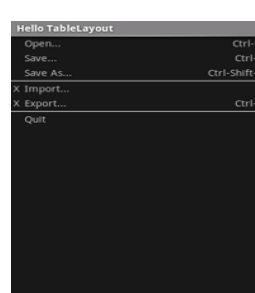
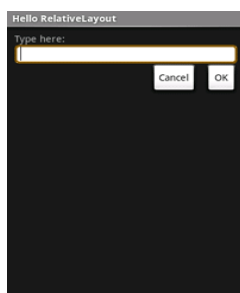
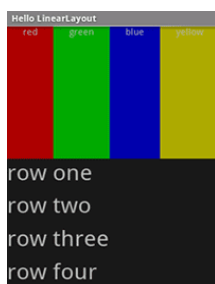
Το LinearLayout αποτελεί διάταξη στοιχείων σε οριζόντια ή κατακόρυφη σειρά. Αν δηλώσουμε στοιχεία μέσα σε ένα οριζόντιο LinearLayout τότε τα στοιχεία αυτά θα εμφανίζονται στην οθόνη σε μία οριζόντια διάταξη με τη σειρά που τα δηλώσαμε το ένα δίπλα στο άλλο.

Το RelativeLayout μας δίνει περισσότερη ελευθερία στη δήλωση των γραφικών στοιχείων, με την έννοια ότι κάθε στοιχείο μπορούμε να το εμφανίσουμε σε συγκεκριμένο σημείο της οθόνης, όπως π.χ. στην αρχή, στο κέντρο, στο τέλος ή να επιλέξουμε τη θέση του σε σχέση με κάποιο άλλο στοιχείο.

Το FrameLayout αποτελεί την απλούστερη διάταξη στοιχείων. Είναι απλά ένας κενός χώρος τον οποίο μπορούμε να γεμίσουμε με κάποιο αντικείμενο, π.χ. μία εικόνα. Για το λόγω αυτό συνήθως χρησιμοποιείται σαν ρίζα (root) στο δέντρο των γραφικών στοιχείων της οθόνης.

Το `TableLayout` αποτελεί διάταξη πίνακα, δηλαδή μπορεί να διατάσσει τα παιδιά του (children) σε σειρές και στήλες. Σε όλα τα παραπάνω στοιχεία μπορούμε να ρυθμίσουμε αρκετές παραμέτρους όπως μέγεθος (πλάτος και ύψος), οριζόντια ή κατακόρυφη διάταξη, βαρύτητα (layout gravity) και άλλες.

Τα `layouts` της εφαρμογής δηλώνονται α) μέσω αρχείων `xml` ή β) μέσα στις `activities` της εφαρμογής. Στις περισσότερες περιπτώσεις δημιουργούμε δυναμικά τα `layouts` μέσα στις `activities` και ορίζουμε εκεί τις παραμέτρους αυτών. Ωστόσο ο συνηθέστερος τρόπος είναι να δημιουργήσουμε για κάθε οθόνη ένα διαφορετικό `xml` αρχείο που θα περιλαμβάνει τη διάταξη όλων των γραφικών της στοιχείων και να το καλέσουμε μέσα από την αντίστοιχη `activity`.



Εικόνα 22: `LinearLayout`[21] Εικόνα23: `RelativeLayout`[22] Εικόνα 24: `TableLayout` [23]
(developer.xamarin.com)

4.8.2 Menu

Τα μενού είναι ένα κοινό συστατικό της διεπαφή χρήστη σε πολλούς τύπους εφαρμογών. Παρέχει μια οικεία και συνεπή εμπειρία χρήστη, πρέπει να χρησιμοποιήσετε τα `API Menu` για να παρουσιάσουμε τις ενέργειες του χρήστη και άλλες επιλογές για τις δραστηριότητές μας. Στο λειτουργικό σύστημα `Android`, υπάρχουν τρία διαφορετικά είδη μενού, το μενού επιλογών (`options menu`), το μενού πλαισίου (`context menu`) και το αναδυόμενο μενού (`submenu`), τα οποία δηλώνονται και αυτά σε αρχεία `xml`.

Το `options menu` αποτελεί το βασικότερο μενού μίας εφαρμογής. Εμφανίζεται τη στιγμή που πατάμε το κουμπί `menu` του κινητού μας τηλεφώνου και περιέχει όλες τις βασικές επιλογές της εφαρμογής μας. Αποτελεί κυρίως τον τρόπο με τον οποίο περιηγούμαστε μεταξύ των διαφορετικών οθονών και

activities της εφαρμογής μας. Στον κώδικα της εφαρμογής μας ορίζουμε κάθε επιλογή του μενού σε ποια activity θα οδηγήσει το χρήστη.

Το context menu ενεργοποιείται από τον χρήστη με παρατεταμένο πάτημα του στοιχείου αυτού. Στο context menu ενός κειμένου θα όριζε κανείς επιλογές “αντιγραφή”, “αποκοπή” κλπ, στο context menu μίας διεύθυνσης URL σε εφαρμογή web browser θα ορίζαμε επιλογές “άνοιγμα”, “άνοιγμα σε νέα καρτέλα” κ.ο.κ.

Ένα popup menu εμφανίζει μια λίστα με στοιχεία σε μια κατακόρυφη λίστα που είναι συνδεδεμένη με την άποψη που επικαλέστηκε το μενού. Δράσεις σε ένα αναδυόμενο μενού δεν θα πρέπει να επηρεάζουν άμεσα το αντίστοιχο περιεχόμενο που είναι ενέργιες για τα συμφραζόμενα. Αντίθετα, το αναδυόμενο μενού είναι για τις εκτεταμένες ενέργειες που σχετίζονται με τις περιοχές του περιεχομένου της activity.

4.8.3 Dialogs

Ένα dialog είναι ένα μικρό παράθυρο που προτρέπει το χρήστη να λάβει μια απόφαση ή να εισάγει επιπρόσθετες πληροφορίες. Ένα παράθυρο διαλόγου δεν γεμίζει την οθόνη και χρησιμοποιείται συνήθως για γεγονότα που απαιτούν από τους χρήστες να αναλάβουν δράση για να μπορέσουν να προχωρήσουν. Τα κυριότερα είδη διαλόγων είναι ο AlertDialog (διάλογος ειδοποίησης) και ο ProgressDialog (διάλογος προόδου).

Ο AlertDialog είναι ο πιο συνηθισμένος διάλογος. Αποτελείται από ένα τίτλο, ένα μήνυμα, ορισμένα κουμπιά ή μία λίστα από επιλογές. Για κάθε κουμπί του διαλόγου, ορίζουμε μέσα στην activity τις ενέργειες που θα ακολουθήσουν όταν το πατήσει ο χρήστης.

Ο ProgressDialog αποτελεί ουσιαστικά επέκταση του AlertDialog και χρησιμοποιείται όταν θέλουμε να εμφανίσουμε στο χρήστη την πρόοδο για κάποια ενέργεια. Για παράδειγμα όταν θέλουμε να κατεβάσουμε κάποιες εικόνες από το διαδίκτυο και να τις εμφανίσουμε στο χρήστη, θα χρειαστούμε κάποιο χρονικό διάστημα για να ολοκληρωθεί αυτή η ενέργεια. Οπότε για να μη βλέπει ο χρήστης μία κενή μαύρη οθόνη, εμφανίζουμε έναν ProgressDialog και στο background εκτελούμε τις χρονοβόρες διαδικασίες.

4.8.4 Notifications

Το Notification είναι ένα μήνυμα που μπορεί να εμφανιστεί στον χρήστη εκτός του κανονικού UI της εφαρμογής. Όταν λέμε στο σύστημα να παράγει μια ειδοποίηση, εμφανίζεται για πρώτη φορά ως εικονίδιο στην περιοχή ειδοποιήσεων. Για να δούμε τις λεπτομέρειες της κοινοποίησης, ανοίγουμε το πλαίσιο ειδοποιήσεων. Τόσο η περιοχή ειδοποιήσεων όσο και ο σχεδιαστής ειδοποιήσεων είναι ελεγχόμενες περιοχές του συστήματος τις οποίες ο χρήστης μπορεί να δει ανά πάσα στιγμή.

Για παράδειγμα όταν ο χρήστης αποθηκεύει ένα αρχείο, θα θέλαμε να δει κάποιο μήνυμα ότι το αρχείο αποθηκεύτηκε επιτυχώς. Ή όταν η εφαρμογή μας τρέχει στο background και θέλει να ενημερώσει το χρήστη για κάποιο γεγονός, θα πρέπει να στείλει κάποια ειδοποίηση την οποία ο χρήστης να μπορεί να “ανοίξει” όταν αυτός επιθυμεί. Στην πρώτη περίπτωση χρησιμοποιούμε toast notification, ενώ στη δεύτερη status bar notification.

4.9 Google Maps

Το Android μας επιτρέπει να ενσωματώσουμε τους Google Maps Android API v2. στην εφαρμογή μας. Μπορούνε να εμφανίσουνε οποιαδήποτε τοποθεσία στο χάρτη, μπορούνε να δείξουνε διαφορετικές διαδρομές στο χάρτη κ.λπ. Μπορούμε επίσης να προσαρμόσουμε το χάρτη σύμφωνα με τις επιλογές μας.

Το Google Maps API είναι δωρεάν για εμπορική χρήση, υπό τον όρο ότι το site/εφαρμογή στην οποία χρησιμοποιείται θα είναι προσβάσιμο στο κοινό χωρίς να χρεώνει τον χρήστη του για κάθε πρόσβαση. Η ερώτηση στο Google Maps API γίνεται με την αποστολή ενός αιτήματος HTTP GET στην Web εφαρμογή, και επιστρέφει την απάντηση σε μορφή XML ή JSON. Η εφαρμογή η οποία περιγράφεται σε αυτή την διπλωματική χρησιμοποιεί μηνύματα JSON.

Η εφαρμογή χρησιμοποιεί την έκδοση του API, το Google Maps Android API v2. Για να χρησιμοποιηθεί το Google Maps API στην εφαρμογή απαιτείται η απόκτηση ενός “κλειδιού” (Google Maps API Key) από τον δημιουργό της εφαρμογής καθώς επίσης και η εισαγωγή του στον κώδικα της εφαρμογής. Για την απόκτηση είναι απαραίτητη η δημιουργία λογαριασμού Google και η εγγραφή στο Google APIs Console (<https://code.google.com/apis/console/>).

Στη συνέχεια για τη χρήση του εργαλείου πρέπει να οριστεί στο xml αρχείο Manifest της εφαρμογής (AndroidManifest.xml) το κλειδί το οποίο έχει δοθεί από την Google με την εισαγωγή του παρακάτω αντικειμένου σαν "παιδί" στοιχείου <application>: <meta-data

```
android:name="com.google.android.maps.v2.API_KEY"
```

```
android:value="your_api_key"/>
```

Για τη χρήση του Google Maps Android API v2 χρειάζεται ο καθορισμός στην εφαρμογή δικαιωμάτων πρόσβασης σε λειτουργίες του συστήματος. Ο καθορισμός ενός δικαιώματος γίνεται με την προσθήκη ενός στοιχείου <uses-permission> σαν παιδί του αντικειμένου <manifest> στο αρχείο AndroidManifest.xml της εφαρμογής.

Τα δικαιώματα που χρειάζεται να καθοριστούν για τη χρήση του Google Maps API v2 είναι:

- android.permission.INTERNET: Χρησιμοποιεί το API για τη λήψη του χάρτη από τους Google Maps servers στη συσκευή.
- android.permission.ACCESS_NETWORK_STATE: Επιτρέπει στο API να ελέγξει τη κατάσταση των συνδέσεων τις συσκευής, έτσι ώστε να καθορίσει αν μπορεί να γίνει λήψη δεδομένων.
- android.permission.WRITE_EXTERNAL_STORAGE: Επιτρέπει στο API να καταχωρίσει τα δεδομένα του χάρτη στον χώρο αποθήκευσης της συσκευής.
- com.google.android.providers.gsf.permission.READ_GSERVICES: Επιτρέπει στο API την πρόσβαση σε web υπηρεσίες της Google .[25][26]

5 Υλοποίηση και λειτουργικότητα της εφαρμογής

5.1 Περιγραφή της εφαρμογής

Για την εκκίνηση της εφαρμογής είναι απαραίτητο η συσκευή μας να είναι συνδεδεμένη στο internet μέσω Wi-fi ή μέσω των δεδομένων κινητής τηλεφωνίας και να έχουμε ενεργοποιήσει το GPS.

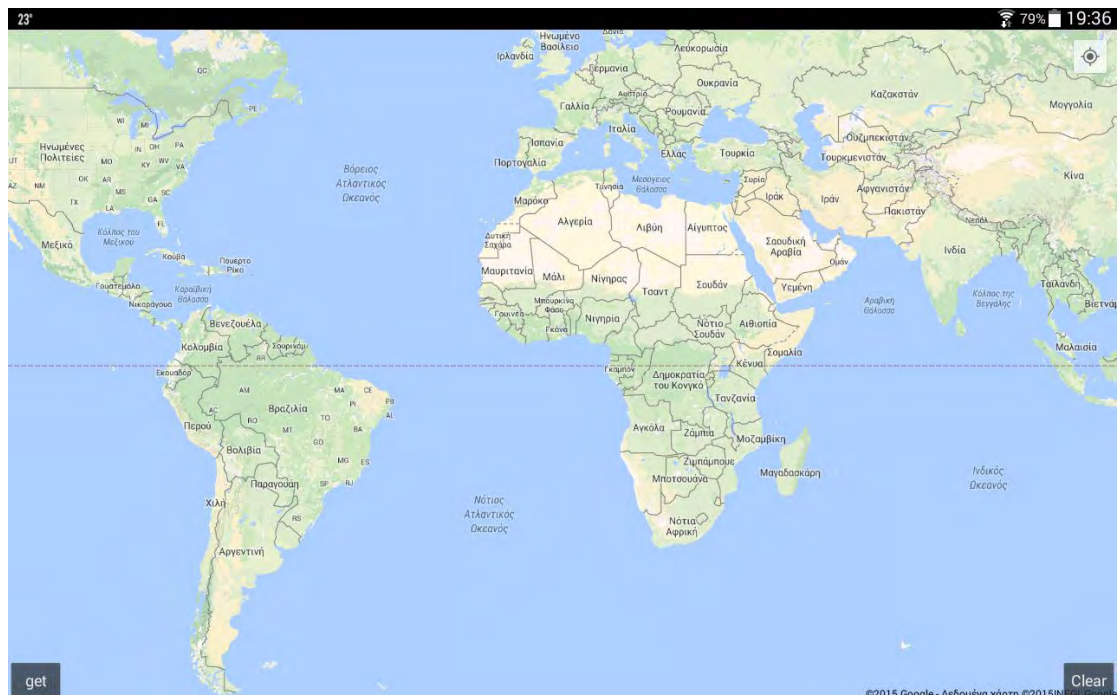
Στην οθόνη της συσκευής μόλις ανοίξει η εφαρμογή εμφανίζεται ο χάρτης της google εφόσον ο έλεγχος ήταν επιτυχής και τα Google Play Services είναι διαθέσιμα, όπως φαίνεται παρακάτω.

```
int status =
GooglePlayServicesUtil.isGooglePlayServicesAvailable(getBaseContext());

if(status!=ConnectionResult.SUCCESS){ // Google Play Services are not
available

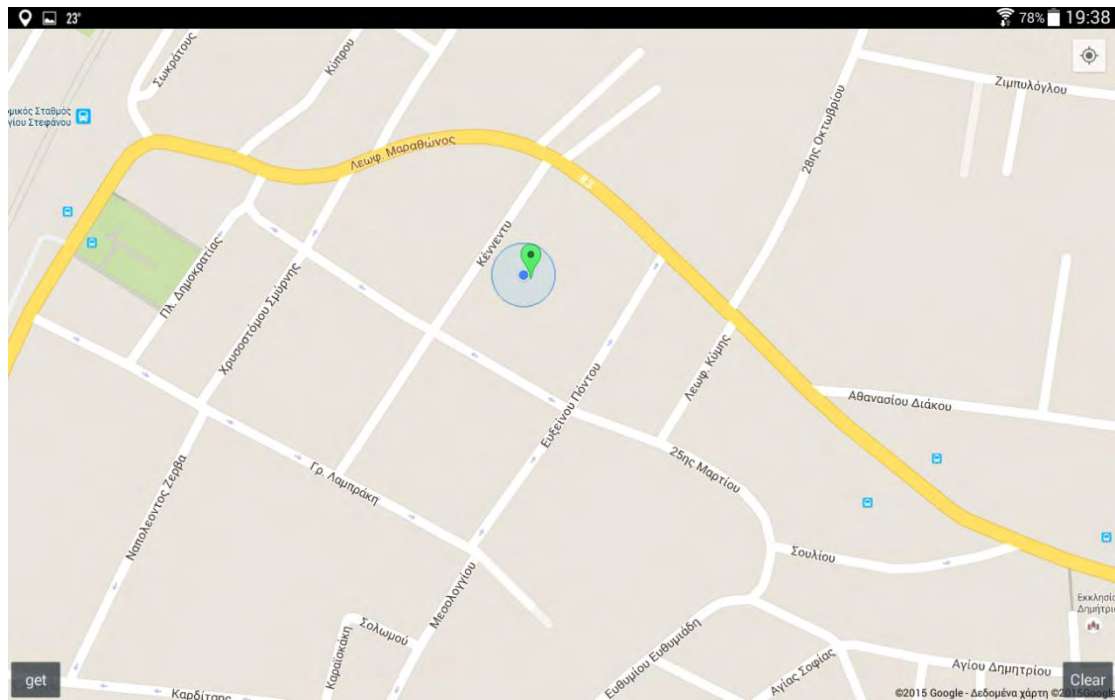
    int requestCode = 10;
    Dialog dialog = GooglePlayServicesUtil.getErrorDialog(status,
this, requestCode);
    dialog.show();

}else { // Google Play Services are available
```



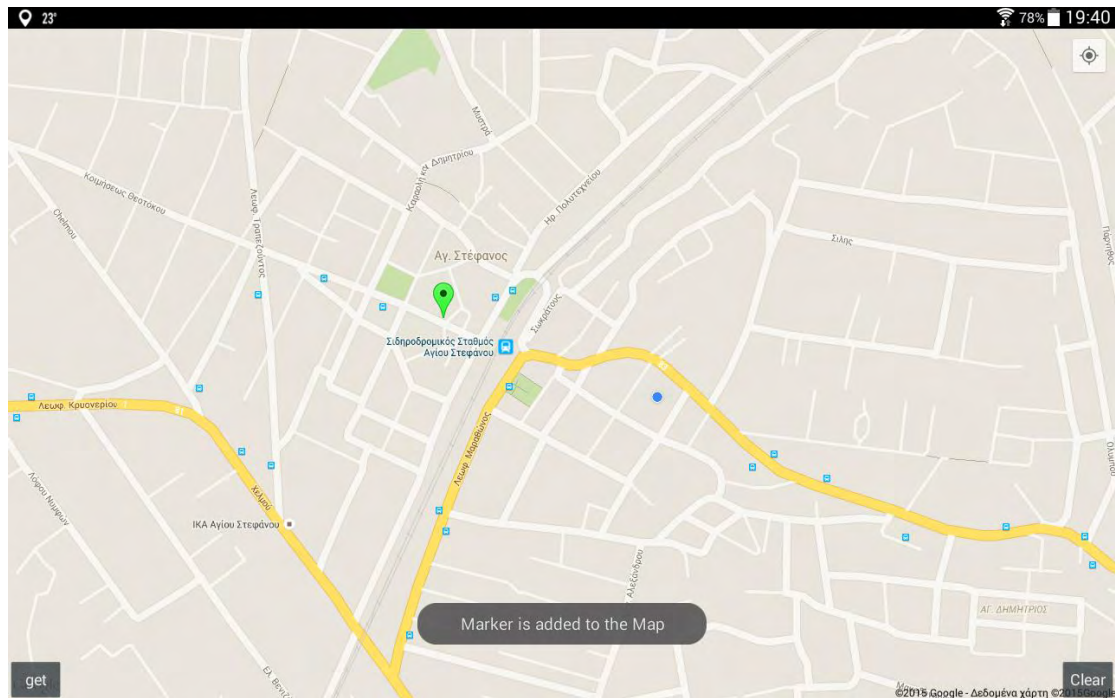
Εικόνα 25: Οθόνη υποδοχής

Εφόσον ο έλεγχος ήταν επιτυχής και οι χάρτες μας είναι διαθέσιμοι ο χρήστης έχει τη δυνατότητα να ενεργοποιήσει το GPS ώστε να προσδιοριστεί η ακριβής του θέση πάνω στο χάρτη. Μόλις αυτό επιτευχθεί ο χρήστης μπορεί να εστιάσει στη θέση του μέσω του κουμπιού που βρίσκεται πάνω και δεξιά στην εφαρμογή.



Εικόνα 26: Στιγμιότυπο προσδιορισμού ακριβής θέσης

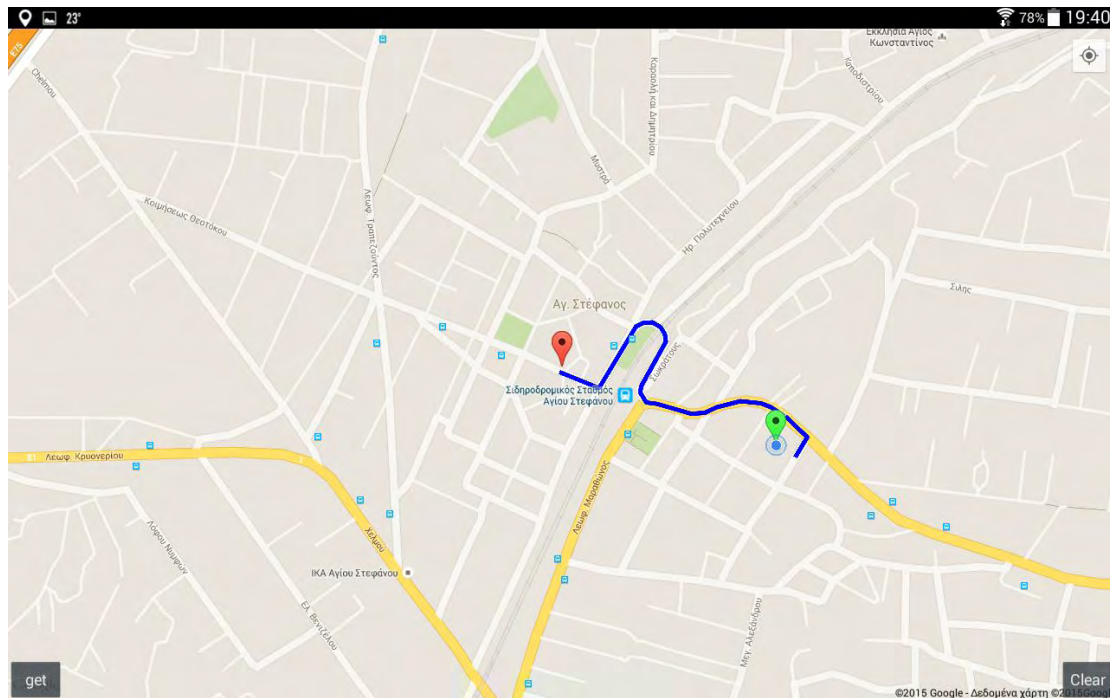
Η μπλε κουκίδα που εμφανίζεται είναι η θέση που βρισκόμαστε. Με τον πράσινο marker που υπάρχει στον χάρτη μας, ο χρήστης μπορεί να αποθηκεύσει στη βάση δεδομένων την ακριβή θέση που στάθμευσε το όχημά του. Αυτή την ενέργεια την κάνει ο χρήστης χειροκίνητα με τη βοήθεια που του παρέχει το GPS. Η επιλογή να μην αποθηκεύεται η θέση του οχήματος αυτόματα έγινε σκοπίμως διότι σε πολλές περιπτώσεις, παρότι ο πράσινος marker βρίσκεται τοποθετημένος εντός της εμβέλειας του GPS δεν είναι στην ακριβή θέση. Οπότε κάνοντας κλικ σε ένα σημείο του χάρτη ο χρήστης έχει αποθηκευμένη την θέση στάθμευσης του όπως φαίνεται στην επόμενη εικόνα.



Εικόνα 27: Στιγμιότυπο αποθήκευσης της θέσης στάθμευσης

Εφόσον έχει αποθηκευτεί η θέση του οχήματος ο χρήστης μπορεί να κλείσει την εφαρμογή.

Όταν ο χρήστης θελήσει να επιστρέψει στη θέση που πάρκαρα το όχημά του δεν έχει παρά να ανοίξει την εφαρμογή. Αμέσως θα προσδιοριστεί η νέα του θέση μέσω του GPS. Στη συνέχεια ο χρήστης πατώντας το κουμπί **get** που βρίσκεται κάτω αριστερά, βλέπει στο χάρτη ποια διαδρομή θα διασχίσει ώστε να φτάσει στο όχημά του. Η σύνδεση γίνεται μέσω του πράσινου marker που βρίσκεται στο εύρος του GPS, δηλαδή της θέσης που είναι εκείνη τη στιγμή ο χρήστης και του κόκκινου marker που αντιπροσωπεύει το τελευταίο σημείο του χάρτη που έχει αποθηκεύσει ο χρήστης στη βάση δεδομένων, δηλαδή της θέσης που είναι σταθμευμένο το όχημά του.



Εικόνα 28: Στιγμιότυπο εμφάνισης της διαδρομής που πρέπει να διασχίσει ο χρήστης

Τέλος με το κουμπί clear που βρίσκεται κάτω δεξιά της εφαρμογής, ο χρήστης μπορεί να σβήσει όλες τις τοποθεσίες του χάρτη που έχει αποθηκεύσει έως εκείνη τη στιγμή στη βάση δεδομένων.

5.2 Δοκιμές εφαρμογής

Οι δοκιμές για να πιστοποιηθεί η σωστή λειτουργία της εφαρμογής πραγματοποιήθηκαν σε δύο πραγματικές συσκευές Android.

Δεν προτιμήθηκε να γίνουν οι δοκιμές σε εικονική συσκευή Android διότι οι απαιτήσεις για το μέγεθος της μνήμης RAM ήταν μεγάλες και δεν επαρκούσε η μνήμη του υπολογιστή στον οποίο έγινε η ανάπτυξη του κώδικα της εφαρμογής. Επίσης, αρκετά άρθρα ανέφεραν ότι είναι προτιμότερο όταν η εφαρμογή διαθέτει χάρτες της Google οι δοκιμές να γίνονται σε πραγματικές συσκευές Android.

Η πρώτη συσκευή ήταν ένα tablet το Samsung Galaxy Tab 3 10.1 P5210 με λειτουργικό Android και έκδοση 4.2.2 (Jelly Bean). Διαθέτει διπύρνηνο επεξεργαστή στα 1.6 GHz και 1GB RAM. Με το tablet η σύνδεση στο διαδίκτυο γινόταν μέσω Wi-Fi.

Η δεύτερη ήταν ένα κινητό το Sony Xperia ion HSPA με λειτουργικό Android και έκδοση 4.4(Kit Kat). Διαθέτει διπύρνηνο επεξεργαστή στα 1.5 GHz και 1GB

RAM. Με το κινητό δοκιμάστηκε η λειτουργία της εφαρμογής όταν η σύνδεση στο διαδίκτυο γινόταν μέσω των δεδομένων κινητής τηλεφωνίας.

Τα αποτελέσματα και από τις δύο συσκευές ήταν τα ίδια χωρίς διαφοροποιήσεις.

5.3 Εξέλιξη της εφαρμογής

Η εξελιγμένη μορφή της εφαρμογής θα καταγράφει μέσω του κινητού την πορεία που ακλουθεί το όχημα που επιβαίνουμε ή οδηγούμε και όταν το όχημα σταματήσει να κινείται θα αποθηκεύεται η περιοχή και ειδικότερα η θέση που στάθμευσε το όχημα μας. Βέβαια επειδή θα κινούμαστε και με τα πόδια είναι προτιμότερο να θέσουμε κάποιο κάτω όριο στην ταχύτητα (π.χ 10χμ ή 20 χμ) για να αποφύγουμε την καταγραφή της διαδρομής που θα ακολουθούμε με τα πόδια (καθώς όταν περπατάμε δεν αναπτύσσουμε τέτοιες ταχύτητες). Βέβαια δημιουργείται ένα πρόβλημα γιατί όταν είναι να σταθμεύσουμε η ταχύτητα μας θα είναι μικρότερη του ορίου που θα έχουμε θέσει, άρα θα μπορούσαμε να κρατήσουμε την ευρύτερη θέση που αναπτύξαμε τέτοια ταχύτητα και να βλέπει ο χρήστης την ευρύτερη περιοχή που έχει σταθμεύσει (πχ οικοδομικό τετράγωνο). Φυσικά δεν θα έχουμε την ακριβή θέση στάθμευσης αλλά όλη η διαδικασία θα γίνεται αυτόματα χωρίς να σημειώνει και να αποθηκεύει τις θέσεις ο ίδιος ο χρήστης.

ΠΑΡΑΡΤΗΜΑ

Δημιουργία βάσης: LocationsDB.java

```
package info.androidhive.googlemapsv2;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class LocationsDB extends SQLiteOpenHelper{

    private static String DBNAME = "locationmarkersqlite";
    private static int VERSION = 1;
    public static final String FIELD_ROW_ID = "_id";
    public static final String FIELD_LAT = "lat";
    public static final String FIELD_LNG = "lng";
    public static final String FIELD_ZOOM = "zom";
    private static final String DATABASE_TABLE = "locations";
    private SQLiteDatabase mDB;

    public LocationsDB(Context context) {
        super(context, DBNAME, null, VERSION);
        this.mDB = getWritableDatabase();
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String sql =
            "create table " + DATABASE_TABLE + " ( " +
            FIELD_ROW_ID + " integer primary key autoincrement , " +
            FIELD_LNG + " double , " +
            FIELD_LAT + " double , " +
            FIELD_ZOOM + " text " +
            " ) ";

        db.execSQL(sql);
    }
    /** Inserts a new location to the table locations */
    public long insert(ContentValues contentValues){
        long rowID = mDB.insert(DATABASE_TABLE, null, contentValues);
        return rowID;
    }
    /** Deletes all locations from the table */
    public int del(){
        int cnt = mDB.delete(DATABASE_TABLE, null , null);
        return cnt;
    }
    /** Returns all the locations from the table */
    public Cursor getAllLocations(){
        return mDB.query(DATABASE_TABLE, new String[] { FIELD_ROW_ID, FIELD_LAT, FIELD_LNG,
            FIELD_ZOOM } , null, null, null, null, null);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }

}
```


Ο Content Provider που κάνει τις λειτουργίες της βάσης:

LocationsContentProvider.java

```
package info.androidhive.googlemapsv2;

import java.sql.SQLException;

import android.content.ContentProvider;
import android.content.ContentUris;
import android.content.ContentValues;
import android.content.UriMatcher;
import android.database.Cursor;
import android.net.Uri;

public class LocationsContentProvider extends ContentProvider{

    public static final String PROVIDER_NAME = "info.androidhive.googlemapsv2.locations";

    public static final Uri CONTENT_URI = Uri.parse("content://" + PROVIDER_NAME +
"/locations" );

    private static final int LOCATIONS = 1;
    private static final UriMatcher uriMatcher ;

    static {
        uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
        uriMatcher.addURI(PROVIDER_NAME, "locations", LOCATIONS);
    }

    LocationsDB mLocationsDB;

    @Override
    public boolean onCreate() {
        mLocationsDB = new LocationsDB(getContext());
        return true;
    }

    @Override
    public Uri insert(Uri uri, ContentValues values) {
        long rowID = mLocationsDB.insert(values);
        Uri _uri=null;
        if(rowID>0){
            _uri = ContentUris.withAppendedId(CONTENT_URI, rowID);
        }else {
            try {
                throw new SQLException("Failed to insert : " + uri);
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return _uri;
    }

    @Override
    public int update(Uri uri, ContentValues values, String selection,
String[] selectionArgs) {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public int delete(Uri uri, String selection, String[] selectionArgs) {
        int cnt = 0;
        cnt = mLocationsDB.del();
        return cnt;
    }

    @Override
    public Cursor query(Uri uri, String[] projection, String selection, String[]
selectionArgs, String sortOrder) {

        if(uriMatcher.match(uri)==LOCATIONS){
            return mLocationsDB.getAllLocations();
        }
        return null;
    }
}
```

```

@Override
public String getType(Uri uri) {
    return null;
}
}

```

Για την εύρεση και τον σχεδιασμό της απόστασης: DirectionsJSONParser.java

```

package info.androidhive.googlemaps2;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import com.google.android.gms.maps.model.LatLng;

public class DirectionsJSONParser {
    public List<List<HashMap<String,String>>> parse(JSONObject jObject){

        List<List<HashMap<String, String>>> routes = new
        ArrayList<List<HashMap<String,String>>>();

        JSONArray jRoutes = null;
        JSONArray jLegs = null;
        JSONArray jSteps = null;

        try {

            jRoutes = jObject.getJSONArray("routes");

            for(int i=0;i<jRoutes.length();i++){
                jLegs = ( (JSONObject)jRoutes.get(i)).getJSONArray("legs");
                List<HashMap<String, String>> path = new ArrayList<HashMap<String, String>>();

                for(int j=0;j<jLegs.length();j++){
                    jSteps = ( (JSONObject)jLegs.get(j)).getJSONArray("steps");

                    for(int k=0;k<jSteps.length();k++){
                        String polyline = "";
                        polyline =
                        (String)((JSONObject)((JSONObject)jSteps.get(k)).get("polyline")).get("points");

                        List<LatLng> list = decodePoly(polyline);
                        for(int l=0;l<list.size();l++){
                            HashMap<String, String> hm = new HashMap<String, String>();
                            hm.put("lat", Double.toString(list.get(l).latitude) );
                            hm.put("lng", Double.toString(list.get(l).longitude) );
                            path.add(hm);
                        }
                    }
                    routes.add(path);
                }
            }

        } catch (JSONException e) {
            e.printStackTrace();
        } catch (Exception e){
        }
        return routes;
    }

    private List<LatLng> decodePoly(String encoded) {

        List<LatLng> poly = new ArrayList<LatLng>();
        int index = 0, len = encoded.length();
        int lat = 0, lng = 0;

```



```

while (index < len) {
    int b, shift = 0, result = 0;
    do {
        b = encoded.charAt(index++) - 63;
        result |= (b & 0x1f) << shift;
        shift += 5;
    } while (b >= 0x20);
    int dlat = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
    lat += dlat;

    shift = 0;
    result = 0;
    do {
        b = encoded.charAt(index++) - 63;
        result |= (b & 0x1f) << shift;
        shift += 5;
    } while (b >= 0x20);
    int dlng = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
    lng += dlng;

    LatLng p = new LatLng((((double) lat / 1E5)),
                           (((double) lng / 1E5)));
    poly.add(p);
}
return poly;
}
}

```

ΕΥΡΗΤΗΡΙΟ ΕΙΚΟΝΩΝ

1. Εικόνα 1: <http://ti-einai.gr/smartphone/>
2. Εικόνα 2: https://el.wikipedia.org/wiki/Λειτουργικό_σύστημα
3. Εικόνα 3: <http://blogs.amdocs.com/interactivities/2011/01/10/direct-billing-a-lucrative-market-for-mobile-operators-worldwide/comment-page-1/>
4. Εικόνα 4: <http://www.droid-life.com/2013/05/14/google-play-updated-to-version-4-1-6-brings-sporty-looking-home-buttons/>
5. Εικόνα 5: <http://androidoriginal.blogspot.gr/p/android-15-cupcake.html>
6. Εικόνα 6: http://logos.wikia.com/wiki/Android_1.6_Donut
7. Εικόνα 7: <https://sites.google.com/site/latestandroidnewscom/android-versions>
8. Εικόνα 8: <http://www.computerworld.com/article/2468294/smartphones/android-2-2-upgrade-list-is-your-phone-getting-froyo.html>
9. Εικόνα 9: <http://bytenow.net/2011/03/22/report-says-android-2-3-gingerbread-will-hit-htc-evo-4g/>
10. Εικόνα 10: <http://phandroid.com/2011/01/26/honeycomb-logo-why-of-course/>
11. Εικόνα 11: <http://www.theinquirer.net/inquirer/feature/2120521/tasty-treats-android-ice-cream-sandwich>
12. Εικόνα 12: <http://www.gadgetfreak.gr/2012/12/18/sony-τα-σχέδια-της-για-jelly-bean-update-και-σε-ποιά-μοντέ/>
13. Εικόνα 13: <http://www.droid-life.com/2013/09/03/kit-kat-will-be-android-4-4-not-android-5-0-download-the-official-wallpaper/>
14. Εικόνα 14: <http://www.webgeometrics.com/android-os-architecture-explained-for-non-techies/>
15. Εικόνα 15: <http://www.edureka.co/blog/wp-content/uploads/2013/01/Android-evolution-Linux-Kernel-versions-1.png>
16. Εικόνα 16: <https://www.flickr.com/photos/12505901@N07/4080191202/>

17. Εικόνα 17: <http://javapapo.blogspot.gr/2012/10/adventures-with-eclipse-juno-macosx.html>
18. Εικόνα 19: http://courses.cs.washington.edu/courses/cse143/11wi/working_at_home.shtml
19. Εικόνα 20: <http://developer.android.com/tools/help/sdk-manager.html>
20. Εικόνα 21: http://www.javasrilankansupport.com/2012_04_01_archive.html
<http://devamigo.com/android-tutorials/create-android-project-in-eclipse/>
21. Εικόνα 22: http://developer.xamarin.com/guides/android/user_interface/
22. Εικόνα 23: http://developer.xamarin.com/guides/android/user_interface/
23. Εικόνα 24: http://developer.xamarin.com/guides/android/user_interface/

ΑΝΑΦΟΡΕΣ – ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Έξυπνα τηλέφωνα <https://el.wikipedia.org/wiki/Smartphone>
2. ΛΣ Android <https://el.wikipedia.org/wiki/Android>
3. Εκδόσεις Android <http://faqoid.com/advisor/android-versions.php#android-timeline>
4. Εκδόσεις Android <http://developer.android.com/about/versions>
5. ΛΣ Android [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
6. Μηχανή Dalvik [http://en.wikipedia.org/wiki/Dalvik_\(software\)](http://en.wikipedia.org/wiki/Dalvik_(software))
7. Αρχιτεκτονική Android <http://www.eazytutz.com/android/android-architecture/>
8. Android tutorial <http://www.tutorialspoint.com/android/index.htm>
9. Android SDK
http://en.wikipedia.org/wiki/Android_software_development#Android_SDK
10. Eclipse [http://en.wikipedia.org/wiki/Eclipse_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software))
11. SDK Manager <http://developer.android.com/tools/help/sdk-manager.html>
12. Android NDK <http://developer.android.com/tools/sdk/ndk/index.html>
13. Android Developer Tools
<http://developer.android.com/tools/help/adt.html>
14. Android database SQLite
<http://developer.android.com/reference/android/database/sqlite/package-summary.html>
15. Android database SQLite
<http://www.vogella.com/tutorials/AndroidSQLite/article.html>

16. Android Manifest
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
17. Android Manifest
<http://www.buzztouch.com/forum/thread.php?tid=C2FCEFDC22B3A9C1ADABE3B¤tPage=1>
18. AsyncTask class
<http://developer.android.com/reference/android/os/AsyncTask.html>
19. AsyncTask class <http://www.compiletimeerror.com/2013/01/why-and-how-to-use-async-task.html#.VXnGvntmkp>
20. Activity class
<http://developer.android.com/guide/components/activities.html>
21. Activity class <http://developer.android.com/training/basics/activity-lifecycle/starting.html>
22. User Interface
<https://developer.android.com/guide/topics/ui/index.html>
23. Resources
<http://developer.android.com/guide/topics/resources/overview.html>
24. Fragments
<http://developer.android.com/guide/components/fragments.html>
25. Android Google Maps <http://www.androidhive.info/2013/08/android-working-with-google-maps-v2/>
26. Android Google Maps
http://www.vogella.com/tutorials/AndroidGoogleMaps/article.html#goglemaps_manifest
27. Services <http://developer.android.com/guide/components/services.html>

